UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

OPEN-SOURCE INTELLIGENCE EM SISTEMAS SIEM

Bernardo de Simas Gaspar Rodrigues

Dissertação orientada pelo Prof. Doutor Alysson Neves Bessani
e co-orientada por Mestre Sérgio Valentim Costa de Sá

DISSERTAÇÃO

MESTRADO EM SEGURANÇA INFORMÁTICA

2015

Agradecimentos

As pessoas que mais obviamente merecem um agradecimento por este trabalho são os meus pais, Adélia e Vitor e o meu irmão, Diogo. Em grande parte devo-lhes o meu sucesso académico e espero poder retribuir, um dia, a vigilância e interesse que demonstraram pelo meu percurso até à data.

Durante a minha passagem pela FCUL conheci pessoas muito interessantes. A chegada a este ponto foi influenciada por vários colegas com os quais fui podendo aprender, mais e melhor, e graças aos quais a faculdade acabou por se tornar uma fase, com princípio e fim. Um abraço para eles, Miguel, Pedro, Rodrigo, Faneca (que será sempre o Faneca), Radu, Mauro, Tiago e André. Espero que em breve possamos trabalhar juntos, de novo.

Um abraço para o Hélio e o Pedro que me ensinaram que, por não ser dotado de uma capacidade intelectual superior, como eles, terei que me superar constantemente.

Obrigado ao Sérgio Sá e a toda a equipa de segurança da Unisys pelo apoio e participação no projecto.

Muito obrigado ao professor Alysson pela paciência e cuidado com que me ajudou a construir este trabalho.

Aos meus pais, Vitor e Adélia,

Ao meu irmão, Diogo.

Resumo

A OSINT é uma interminável fonte de informação valiosa, em qualquer que seja o contexto, no qual exista a necessidade de lidar com ameaças humanas e imprevisíveis. A segurança informática não é excepção a esta regra e o uso de informação proveniente de canais OSINT tem-se, como temos vindo a observar com o advento da *Threat Intelligence*, firmado como um componente fundamental. Propomo-nos, com este trabalho, a integrar este canal de valioso conhecimento no SIEM (um paradigma também indispensável da área) de uma forma automatizada, através de uma ferramenta/*framework* que visa estabelecer a fundação de um instrumento extensível para recolher e reduzir grandes quantidades de informação a conjuntos, utilizáveis e úteis, de valiosos dados e conhecimentos sobre ameaças. Essa ferramenta irá recolher dados e, servindo-se de uma técnica simplista de aprendizagem de máquinas supervisionada, refiná-los, garantindo que ao SIEM apenas é passada informação relevante. Por forma a validar os nossos esforços, providenciamos provas empíricas da aplicabilidade da nossa solução, em contexto prático e real, demonstrando, efectivamente, o poder de síntese, com base em *feedback* do utilizador, da nossa solução. Os nossos resultados apresentam bons indicadores de que a nossa abordagem é viável e que o nosso componente é capaz de reduzir e filtrar volumes significativos de informação de redes sociais a conjuntos, manuseáveis, de informação estratégica.

Palavras-chave: SIEM, OSINT, Intelligence, ArcSight, Computer Security

Abstract

OSINT is a source of endlessly valuable information for all contexts that have to deal with unpredictable human threats. Computer Security is no exception to this idea and the use of OSINT for Threat Intelligence has been widely established as a fundamental component. We propose to integrate this channel of knowledge into a SIEM platform, a widely employed paradigm in the sector. We also aim to do it in an automated fashion, through a tool that tries to lay the groundwork of an extendable instrument to collect and reduce vast amounts of information to usable amounts of threat data. This tool retrieves data and, leveraging a simplistic supervised machine learning technique, refines it ensuring that the SIEM platform is to receive only relevant information. In order to validate our efforts we provide empirical evidence of the applicability of our solution, demonstrating, in practical context, its power for synthesizing information based on user-provided feedback. Our results reveal good evidence that our approach is a viable one and that our prototype is capable of reducing and filtering large volumes of social networking data, to manageable sets of intelligence.

Keywords: SIEM, OSINT, Intelligence, ArcSight, Computer Security

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A evolução e massificação da Internet veio aumentar exponencialmente a rapidez de transferência de informação para qualquer parte do globo, a qualquer instante no tempo. Com o aumento, em igual proporção, da utilização de tecnologia e da Internet em ambientes tanto domésticos como empresariais, e com a crescente dependência que tanto os negócios como as pessoas têm vindo a apresentar face à realidade e aos equipamentos digitais, a exposição de todos estes intervenientes aos riscos e perigos que são inerentes a estes meios torna-se, significativamente, mais evidente.

De um ponto de vista empresarial, as organizações estão cada vez mais sujeitas e sensíveis àquilo que é o crime informático, não só por ser a Internet um meio tão desejável para cometer crime (em que os aspectos jurídicos do mesmo acabam por desvanecer por entre tecnicalidades de todo o tipo), mas também pelo grau de exposição das organizações que se "movimentam" na Internet (que apesar de facilitar algumas actividades do negócio, aumentando a produtividade e a rentabilidade do mesmo, tornam a empresa mais vulnerável e mais exposta ao atacante externo). É também pelo resultado que desses crimes pode surgir que esta temática toma um papel cada vez maior no dia-a-dia da gestão de risco das organizações.

Outras preocupações têm vindo a intensificar-se nos últimos tempos, com o aumento da paranóia e sensibilidade para as temáticas que rodeiam a segurança informática. Acontecimentos como a recente descoberta das graves violações de privacidade conduzidas por um grande organismo Norte-americano no espectro digital ou eventos associados ao "Hacktivismo", levados a cabo por grupos como os "Anonymous" e outros grupos de activistas, são exemplos paradigmáticos que justificam essa crescente preocupação.

Com toda esta panóplia de preocupações, as organizações começam a ver como principal ponto de protecção do negócio, o investimento em medidas de protecção da sua infra-estrutura e dos seus dados e a constante monitorização de ocorrências relevantes do ponto de vista da segurança informática. Entre as várias ferramentas existentes no

mercado, as ferramentas SIEM (Security Information and Event Management) [1] merecem especial destaque pelas suas capacidades de agregar, consolidar, analisar e correlacionar, em tempo-real, eventos e notificações de segurança de todos (ou de uma grande parte) os componentes da infra-estrutura informática de uma empresa.

De um modo geral, todo o tipo de informação de segurança relevante passa por estas ferramentas e é sobre elas que assenta grande parte do trabalho de análise e detecção de ameaças e problemas de segurança.

O ArcSight [2] é uma ferramenta de SIEM desenvolvida pela empresa HP. Trata-se de uma ferramenta de elevada complexidade, capaz de providenciar uma recolha e análise, em tempo-real, de grandes volumes de dados sobre as mais variadas fontes que podem estar situadas por toda a infra-estrutura da empresa. Apesar do seu elevado nível de customização e da versatilidade em termos dos dados que pode receber, a plataforma (e os SIEMs em geral) apresenta uma lacuna, muito frequentemente colmatada pela intervenção humana. É ela a integração, na análise, de informação proveniente de fontes de OSINT (Open-Source Intelligence).

OSINT [3] é o conceito que representa toda a informação[1] que circula em meios abertos e que é de livre acesso por qualquer indivíduo ou organização. Este conceito é muito abrangente e pode englobar os mais variados canais de livre circulação de informação, desde a internet e as emissoras de rádio, até meios menos tecnológicos como a imprensa.

Pela riqueza, variedade e importância da informação que circula dentro deste paradigma (o OSINT), as organizações começam a reconhecê-lo (especialmente as mais sensíveis às temáticas da segurança informática) como um ponto fulcral da actividade de monitorização de segurança. Vemos evidências deste facto em SOCs (*Security Operations Centres* - Centros de Operações de Segurança) modernos, em que os analistas de segurança estão também responsáveis por monitorizar a actividade de ameaças externas nos canais alternativos, acima mencionados. Por norma esta análise está mais focada nos meios, mais comuns, de difusão de informações existentes na internet, como fóruns e redes sociais, mas pode ser expandida aos mais variados meios, como vemos em projectos de grande escala, como o programa ROSIDS [4] (Rapid Open Source Intelligence Deployment System).

O facto de esta ser levada a cabo por indivíduos e não por máquinas (devido, maioritariamente à complexidade inerente à compreensão de texto e aferência de conclusões sobre os conteúdos analisados) acresce à dificuldade de um conjunto de

[1] Entende-se "informação" como sendo a tradução da palavra "*intelligence*", ou seja, informação crítica, estratégica e relevante para a actividade desempenhada por uma empresa ou organização.

tarefas de monitorização (por analistas) que são, já por si, complicadas e podem até chegar ao cúmulo de obrigar à alocação de profissionais especializados.

Com advento do Big Data e o aumento de visibilidade sobre as técnicas de ***data mining*** (extracção de dados) levadas a cabo pelos gigantes corporativos do mundo tenológico (como a Google ou a Facebook), esta possibilidade começa a tomar forma e vemos surgir alguns produtos de CRM, por exemplo, que utilizam técnicas semelhantes para alertar, automaticamente, sobre manifestações de descontentamento por parte de clientes, em redes sociais.

Apesar das vantagens inerentes ao seu uso, a informação proveniente de redes sociais e da internet tende a ser bastante genérica e, no nosso caso específico, necessita de um tratamento prévio que filtre aquilo que é irrelevante (grande parte da informação). Surge, por isso mesmo, a necessidade de abordar o problema indicado e propor uma solução que consiga reduzir os volumes de informação dessas fontes e extrair utilidade sob a forma de informação relevante para a detecção e prevenção de ameaças através dos sistemas SIEM.

1.1 Motivação

O projecto em seguida apresentado insere-se numa parceria estabelecida entre a Faculdade de Ciências da Universidade de Lisboa (FCUL) e a empresa Unisys Portugal. Esta dinâmica, para além de dar aos alunos envolvidos a oportunidade de serem incorporados no contexto empresarial, colocando-os face-a-face com clientes, desempenhando tarefas, em regime de tempo integral, permite que os mesmos desenvolvam um projecto de investigação (como é o caso daquilo que aqui se apresenta).

Esta sinergia entre o mercado empresarial e o ambiente educativo constitui uma excelente oportunidade, permitindo que os formandos comecem a ganhar uma percepção sobre aquilo que se segue (profissionalmente), mantendo, em parte, a sua ligação à componente académica da segurança informática. A possibilidade de interagir com tecnologia amplamente utilizada pelo mercado profissional é também um pilar do que torna esta oportunidade uma excelente abordagem para juntar conhecimento dos "dois mundos".

Especificamente neste projecto o aluno esteve envolvido nas operações da empresa Unisys, apoiando e fornecendo serviços de manutenção, administração e consultoria sobre a plataforma HP ArcSight. A interacção com o cliente é um dado e os participantes são geralmente colocados a trabalhar junto dos clientes, após um período inicial de formação e ambientação, manuseando a plataforma referida.

No contexto específico deste projecto o objectivo foi de casar as duas realidades, produzindo um trabalho académico de relevância empresarial. Aquando da sua realização, e segundo a procura conduzida por nós, não existem produtos ou trabalhos de especificidade suficiente que nos façam crer que o tema de integrar, de forma automatizada, informação OSINT nas plataformas SIEM tenha sido desenvolvido ao mesmo ponto que este trabalho o leva.

1.2 Objectivos

Com o que aqui se apresenta, o nosso foco é o de criar uma ferramenta capaz de prover uma certa facilidade na integração do "canal OSINT" escolhido, na plataforma HP ArcSight, fornecendo os meios necessários para recolher, agregar, armazenar, analisar e transformar a informação que nele flui em conhecimento estratégico para utilização na melhoria da detecção, análise e resposta a ameaças.

Os seguintes tópicos especificam os objectivos que definimos para este trabalho:

1. Estudar a integração de informação OSINT em sistemas SIEM e todos os factores que a influenciam: OSINT, SIEM, ArcSight, cibesegurança e a actividade de centros SOC;
2. Desenvolver uma solução protótipo da ferramenta que idealizámos para resolver o problema de recolher informação OSINT, refiná-la para extrair o maior valor da menor quantidade possível de informação, e transportá-la para um SIEM;
3. Validar o protótipo desenvolvido através de uma prova de conceito que conceda evidências práticas da aplicabilidade do que propomos.

1.3 Contribuições

Com o trabalho aqui presente criou-se uma *framework*, constituindo um ponto de partida no paradigma da integração de informação de canais OSINT, tratada e triada de forma automática, no SIEM ArcSight.

Este esforço constitui uma base de partida para possíveis trabalhos vindouros que poderão melhorar os vários "pilares" da solução (recolha de informação, processamento de relevância e integração com o SIEM) ou introduzir novas funcionalidades.

Para além do trabalho de investigação construído ao longo deste processo, o resultado mais evidente que aqui se apresenta é o protótipo por nós construído. Esta ferramenta que desenvolvemos retira informação da rede social Twitter e refina-a, através de uma técnica simplista de aprendizagem de máquinas supervisionada, passando depois a normalizá-la e enviá-la para o SIEM ArcSight para que possa ser utilizada pelos analistas afectos às tarefas de monitorização de segurança.

Não foi nossa preocupação inicial a utilização de uma técnica de aprendizagem de máquinas para refinamento automático dos conteúdos extraídos pela ferramenta do canal OSINT. Apesar disso, ao longo do trabalho apercebemo-nos de que tal actividade seria necessária, pela natureza da informação que extraímos das redes sociais (que estão repletas de conteúdo inútil, da perspectiva do analista de segurança). O foco deste trabalho não é, no entanto, a aprendizagem de máquinas.

1.4 Estrutura do documento

Este documento está organizado da seguinte forma:

- Capítulo 2 – Introdução dos conceitos basilares da nossa solução. Aqui explicamos os conceitos base como o SIEM, a OSINT e a plataforma ArcSight (a sua função, capacidades e componentes) e apresentamos a nossa proposta teórica para lidar com o problema-alvo do trabalho. Adicionalmente, são apresentados trabalhos relacionados com este e outra investigação e produtos relevantes sobre as áreas da periferia que acabaram por afectar as nossas decisões durante o decorrer do trabalho, tais como a **extracção de dados**, os usos da OSINT noutros ambientes e até a importância da informação estratégica no contexto da segurança informática;
- Capítulo 3 – Detalhamos o funcionamento da nossa solução, em termos da arquitectura pela qual optámos e dos vários componentes que a constituem (*software* do qual nos servimos, *scripts* desenvolvidos e decisões de desenho);
- Capítulo 4 – Apresentamos a nossa prova de conceito, demonstrando a avaliação que fizemos com base num cenário real e com informação real;
- Capítulo 5 – Descrevemos o trabalho e as conclusões que do mesmo podemos retirar, discutimos o conteúdo e os resultados apresentados e especulamos sobre a possibilidade de trabalhos futuros que deste possam surgir;

Capítulo 2

SIEMs e OSINT

Nesta secção introduzem-se os vários conceitos essenciais para a compreensão da solução proposta e das decisões tomadas na sua construção. Apesar de não serem apresentados em grande detalhe, explicamos os paradigmas de SIEM e OSINT e detalhamos sobre o funcionamento da plataforma ArcSight. Por fim, explicitamos a base teórica da nossa proposta para resolver o problema de integrar, de forma automatizada, informação estratégica de canais OSINT nos sistemas SIEM.

2.1 *Security Information and Event Management* (SIEM)

Os sistemas SIEM são uma tendência relativamente recente no sector empresarial das TI, tendo a sua definição sido cunhada em 2005, quando dois investigadores da organização Gartner, de nomes Mark Nicolett e Amrit Williams, os descreveram como sendo plataformas capazes de recolher, analisar, armazenar e apresentar informação, sobre dados colectados através de uma infra-estrutura de IT, a partir de vários dispositivos como IDSs (*Intrusion Detection Systems*), *Firewalls*, IAMs (*Identity Access Managers*), entre outros. O termo tem origem em dois conceitos distintos, que, aquando da sua criação, visavam colmatar diferentes necessidades: SIMs (sistemas *Security Information Management*, ou de Gestão de Informação de Segurança), focados na capacidade de análise em tempo real, com vista a facilitar a Resposta a Incidentes; e SEMs (sistemas *Security Event Management*, ou de Gestão de Eventos de Segurança), orientados principalmente ao armazenamento de longa duração, e análise histórica de eventos, apoiando, por exemplo, actividades de análise forense conduzidas em auditorias de conformidade. [5]

A convergência das duas abordagens surge mais tarde, incentivada pelas necessidades do mercado: com o desejo, por parte dos auditores, de retirar maior partido do SIM e com a "conformidade" a tornar-se um subsector cada vez mais relevante e desejado, com excelentes perspectivas de produção de receitas. [5, 6]

O SIEM acaba por se transformar, então, numa combinação destas plataformas, cumprindo (tão bem quanto a tecnologia e a qualidade do levantamento de requisitos permitem) funções de ambas. Trata-se de uma poderosa ferramenta, construída para facilitar as seguintes funcionalidades/funções [1]:

- **Recolha e Gestão de *logs*** - recolhe dados de *logs* de várias fontes, armazenando e mantendo a informação, consoante a necessidade;

- **Correlação de Eventos** – utiliza e relaciona dados de eventos para descobrir evidências de actividades suspeitas e produzir conteúdo, como relatórios e alertas, de forma automatizada (e possivelmente em tempo-real);

- **Conformidade Legislativa de TI** – é amplamente utilizado como meio de aferir a conformidade (e não-conformidade), através de regras, relatórios e outros conteúdos oferecidos;

- **Resposta Activa** – pode prover funcionalidades para despoletar certos mecanismos de resposta automática, possivelmente melhorando o comportamento e/ou modificando a exposição de dispositivos (existem, naturalmente, vários problemas que podem advir desta abordagem);

- **Segurança de *endpoints*** – pode monitorizar a segurança de múltiplos *endpoints* de uma forma centralizada, facilitando, a tarefa de controlar/aferir a "saúde" de um sistema.

No que toca à sua arquitectura, os SIEMs seguem, geralmente, uma estrutura simplista e linear, em que a informação flui das fontes até ao utilizador final e/ou a um componente de armazenamento. A figura 1 esquematiza a organização da mesma, cujos componentes fundamentais são descritos a seguir.

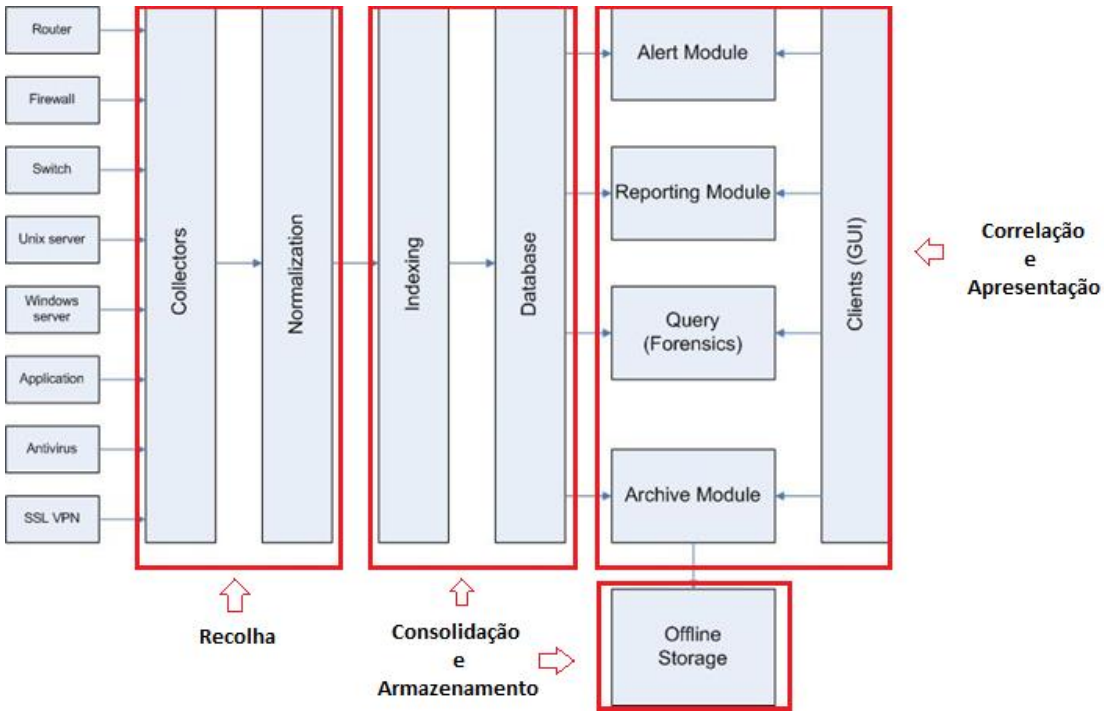

Figura 1:SIEM-Architecture [7]

O mecanismo de **recolha** engloba todas as actividades associadas à colecta e normalização de *logs* (ilustradas na figura como as duas actividades, intituladas "*Collectors*" e "*Normalization*"). **Agentes** pertencentes à plataforma **recolhem** dados de eventos das fontes mencionadas, **normalizando-os** (para uso por parte dos restantes componentes) em referência a um formato bem definido e reconhecido por toda a plataforma, e **passando-os** ao seguinte componente que trata a consolidação dos dados.

Em seguida o mecanismo de **consolidação e armazenamento** manuseia os dados recebidos, **guardando-os** numa base de dados centralizada, segura e robusta. Este componente pode, também, estar encarregue de gerir outras funcionalidades mais complexas como a **retenção** de dados de **longa data**, em locais geograficamente dispersos, por exemplo.

O acesso a esta base de dados de eventos é, então, facilitado ao mecanismo final de **correlação e apresentação** que irá, por sua vez, **operar** sobre os dados existentes, **correlacionando** eventos (ou conjuntos de eventos) e **produzir** informação relevante, tal como: relatórios, alertas e instrumentos de visualização de dados em tempo-real, para "consumo" pelo utilizador final. A informação gerada por este "plano" varia consoante as várias plataformas e, geralmente, os produtos SIEM permitem a criação, por parte dos seus utilizadores, de conteúdo, abrindo as portas à criação de conhecimento estratégico, orientado às necessidades particulares de cada contexto específico. Nesta parte do sistema podemos também ver componentes que provêem a possibilidade de automatizar a resposta a incidentes. [1]

Tendo em conta que uma grande parte da significância deste paradigma está associada à informação que ele é capaz, tanto de produzir como de apresentar, ao utilizador, podemos entender o potencial do SIEM, observando a sua utilização em Centros de Operações de (Ciber)Segurança (CSOC ou SOC) modernos.

2.2 HP ArcSight

O ArcSight [2] é o SIEM produzido e distribuído pela empresa Hewlett-Packard (HP). Trata-se de uma plataforma complexa que engloba todas as funções de um SIEM que foram previamente descritas. É esta ferramenta que vai servir de base para a nossa análise e é precisamente para aumentar as capacidades da mesma que desenvolvemos a solução proposta, sustentando nela a nossa prova de conceito.

Esta plataforma, como veremos, é de uma natureza bastante complexa, devido à flexibilidade que oferece em termos de funcionalidade e equipamento utilizado. O

ArcSight é constituído por vários componentes que formam o sistema base, sobre o qual podem ser adicionadas múltiplas outras "peças", elevando as capacidades do mesmo.

Esta secção descreve a estrutura base da plataforma, bem como alguns dos componentes e recursos que são colocados à nossa disposição pelo fabricante. Pretendemos enquadrar a solução no contexto do trabalho, dando algumas noções essenciais que serão úteis para compreender a interacção entre a ferramenta por nós construída e o ArcSight.

2.2.1 Arquitectura

A figura 3 esquematiza a arquitectura do ArcSight, tal como subdivida entre os vários mecanismos que acima explicámos como constituintes dos sistemas SIEM.

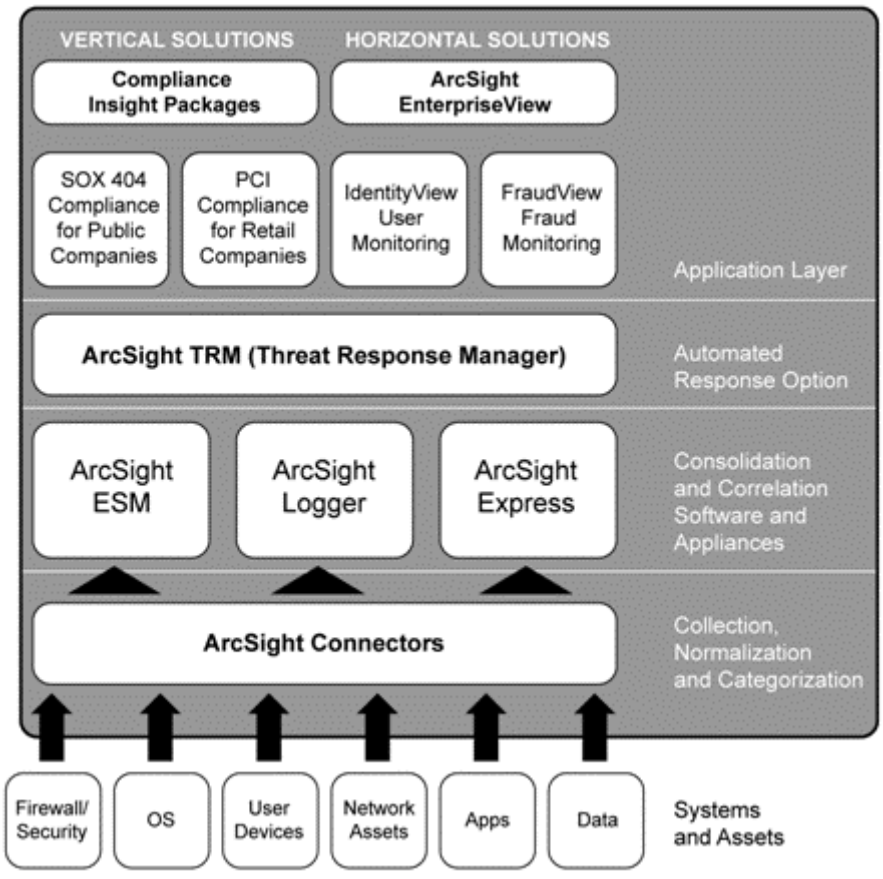

Figura 2: Arcsight architecture [8]

Como podemos observar, apesar de pequenas diferenças no detalhe, a estrutura da arquitectura do sistema ArcSight vai ao encontro daquilo que foi explicitado em relação à generalidade dos sistemas SIEM.

Neste contexto específico, o plano de **recolha** compreende os "*Connectors*", que são componentes isolados que o utilizador/administrador instala numa máquina (que pode ser uma plataforma com múltiplas finalidades como um servidor Windows ou um equipamento específico do ArcSight, designado *Connector Appliance*) e configura, para que colecte dados de *logs* de dispositivos ou aplicações, já existentes.

Seguidamente os 3 componentes denominados "ESM", "*Logger*" e "*Express*" controlam os processos de **consolidação e armazenamento**, lidando com todas as actividades associadas a esse mecanismo.

Os restantes componentes constituem aquilo que definimos como sendo o plano de **correlação e apresentação** e que pode conter vários constituintes, desde ferramentas para construir artefactos visuais de análise de informação, a pacotes de conteúdo com regras e aletas, para aferir e validar características de conformidade legal.

Alguns dos componentes que vão sendo mencionados podem ser opcionais ou distribuídos separadamente da solução principal que é, em suma, composta por: conectores ("*Connectors*") e um equipamento de gestão, correlação e apresentação, centralizadas, designado ESM/*Express* (apresentado mais adiante).

2.2.2 Componentes

A secção seguinte apresenta resumidamente os vários componentes que podem constituir uma instalação regular da solução.

2.2.2.1 ArcSight *Connectors*

Os conectores ("*Connectors*") ArcSight são a "unidade fundamental" da plataforma, já que são eles os componentes que recolhem e encaminham os dados. Encarregam-se de normalizar a informação das fontes (que o utilizador decide incluir na análise) em referência a um formato específico à plataforma, designado CEF (*Common Event Format*). São geralmente específicos a um determinado dispositivo ou fabricante, isto é, geralmente, existe uma instalação específica para normalizar e recolher informação de um determinado ponto. Existem dois principais tipos de conectores [8]:

- ***Smart Connectors***
    - Produtos de *software* "*off-the-shelf*" concebidos pela equipa da HP ArcSight para integrar fontes de dados específicas. Alguns *exemplos* incluem:

        - "**Smart Connector for Oracle Audit DB**" (que, como o nome sugere, recolhe dados de DBs Oracle)
        - "**Smart Connector for IBM System Log for z/OS File**" (utilizado para recolher dados de equipamentos z/OS da IBM, utilizando o suporte de ficheiros para o efeito)
- ***Flex Connectors***
    - Conectores desenhados à medida e que podem ser desenvolvidos quer pela equipa HP ArcSight quer pela equipa responsável pela implementação da solução. São pequenas aplicações que realizam *parsing* sobre informação contida em registos e *logs* fornecidos e a mapeiam para os campos do formato CEF para, mais tarde, os passarem aos componentes de **correlação e armazenamento**.

As soluções ArcSight englobam, normalmente, uma mistura destes dois tipos de conectores, já que as organizações tendem a utilizar um conjunto de aplicações e/ou dispositivos mais diversificado que a oferta de *Smart Connectors* já existente (disponibilizada aos clientes do fabricante). Questões associadas às versões do *software* e às características regionais, como linguagem e fuso horário, são também atributos que podem complicar a tarefa de implementação/configuração da solução, pela entropia que inserem no sistema quando não são contemplados.

2.2.2.2 ArcSight *Connector Appliance*

A *Connector Appliance* é um produto de *software* vendido, por norma, em conjunto com um equipamento de *hardware* e cujo foco é o de gerir conectores, facultando uma plataforma centralizada, simplificada e publicada como uma aplicação *web*. Estes dispositivos são amplamente utilizados em ambientes que requerem gestão de um vasto número de conectores e/ou outras *Connector Appliances*, já que permitem a realização de alterações direccionadas ou sobre conjuntos de conectores controlados remotamente ou instalados no próprio dispositivo. [8]

2.2.2.3 ArcSight *Logger*

O *Logger* é uma plataforma de *storage*, orientada a providenciar, com elevado desempenho, armazenamento de longa data numa base de dados segura e difícil de

manipular/sabotar. Este componente é especialmente útil para manter e assegurar conformidade, já que consegue comportar grandes quantidades de informação comprimida, ainda que dando ao utilizador a possibilidade de extrair eventos originais e não-modificados, de forma trivial e directa.

Em última análise, pode ser utilizado como meio de sustentar infra-estruturas/instalações com elevadas taxas de débito de eventos, graças às funcionalidades de que dispõe para efeitos de escalabilidade. Tem a capacidade de servir, por exemplo, como um intermediário de eventos, recebendo-os dos conectores e propagando-os para um ESM ou *Express* [8].

2.2.2.4 ArcSight *Express* ou ESM (*Enterprise Security Manager*)

Apesar de possuírem diferentes nomes, os dois dispositivos servem, essencialmente, o mesmo propósito, constituindo o núcleo da solução ArcSight. A principal diferença entre os mesmos reside na escala para que cada um é concebido, sendo o *Express* mais orientado a operações de média a pequena dimensão, e o ESM direccionado a operações de grande escala, ou superior (grandes volumes de informação em ambientes multifacetados e bastantes activos). O primeiro contém grande parte das funcionalidades do segundo, apesar de não conseguir alcançar o mesmo desempenho deste.

O dito "núcleo" está, então, dividido em 3 principais componentes, que compõem o plano de gestão e análise: o *Manager*, a base de armazenamento "*CORR-Engine"* e as interfaces de utilizador disponibilizadas.

O *Manager* é um servidor (aplicação) Java que oferece todos os serviços esperados de uma solução SIEM. Este contém toda a funcionalidade associada às actividades de armazenar (interagindo com o seu correspondente de armazenamento, a *CORR-Engine*) e correlacionar eventos, gerindo os mecanismos de *workflow* e comunicação, e facilitando a análise através de conteúdo base (disponível à partida na plataforma) ou desenvolvido à medida, como regras, alertas, relatórios e artefactos de visualização ("*dashboards*").

A infra-estrutura de armazenamento, *CORR-Engine*, por sua vez, recebe dados de eventos para armazenamento e é, em termos simples, uma base de dados de elevado desempenho para indexação, processamento e recolha.

Finalmente as interfaces de utilizador disponibilizadas dividem-se em 3 principais ferramentas para permitir configuração, administração e visualização de dados dentro da plataforma: as "consolas" ArcSight são aplicações para postos-de-trabalho orientadas ao uso por parte do analista, que garantem funcionalidade para criar conteúdo e visualizá-lo; a ArcSight *Management Console* é uma interface *web* focada, essencialmente, em

disponibilizar funções administrativas, como a capacidade de adicionar e remover utilizadores, actualizar licenças de *software* e configurar características de alerta automáticas; por último, a consola *Web* do ArcSight é outra interface *web* que visa expor as capacidades básicas da consola regular através de uma aplicação web, de uma forma simples, segura e acessível através de um *browser* [8].

A figura 3 dá-nos uma percepção de como os dados, por norma, fluem através dos vários componentes da solução; a figura demonstra também um pouco da funcionalidade de como eles interagem para cumprir/oferecer as tarefas/capacidades de um SIEM. Note-se que, apesar de separadas, as catalogações (*labels*) 2 e 3 ("*Event priority evaluation and asset model lookup*" e "*Correlation*") são parte das plataformas ESM/*Express*, acima mencionadas.

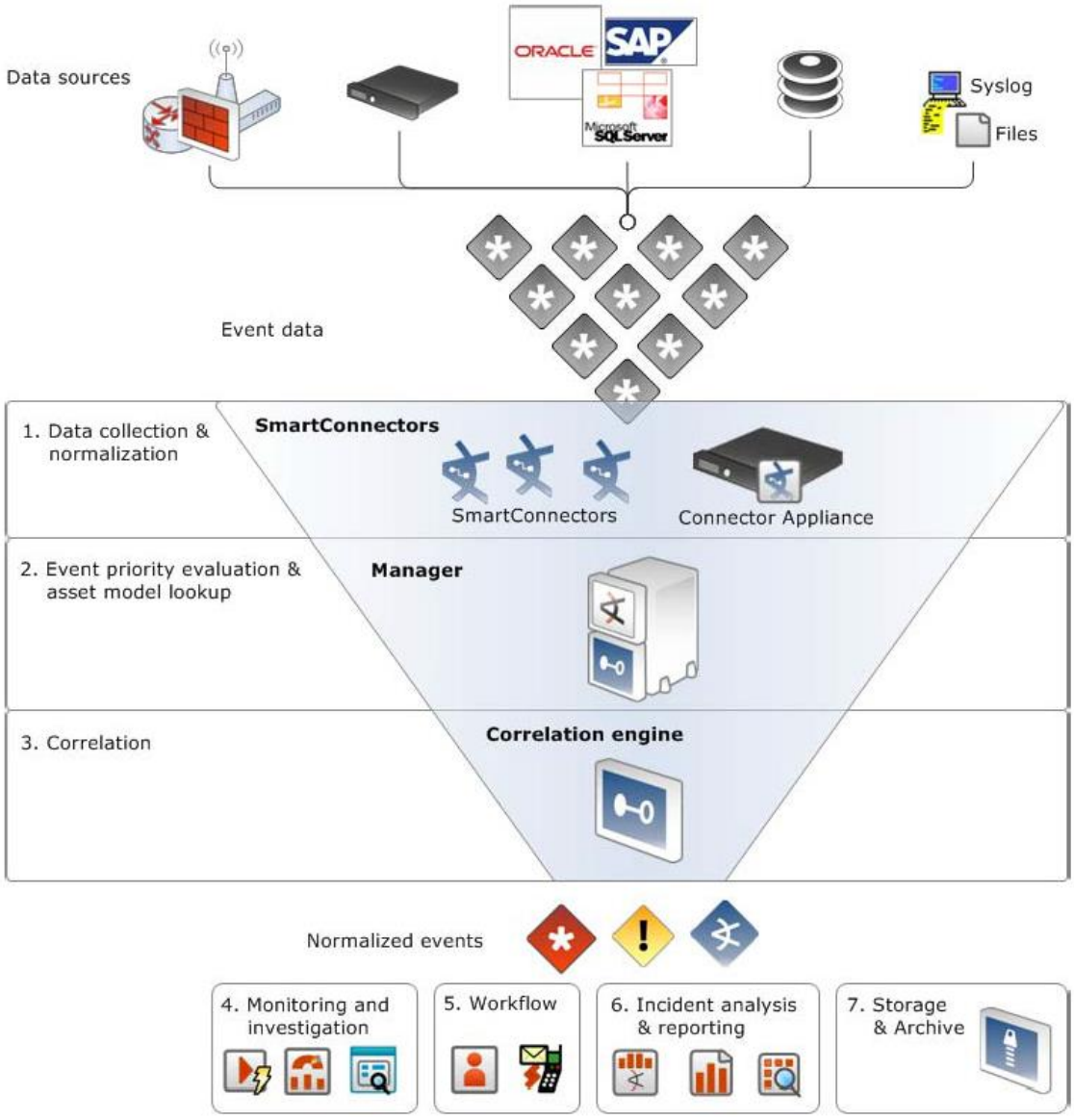

Figura 3: ArcSight Information Flow [2]

2.2.2.5 Outros componentes

Há ainda várias outras opções/adições que podem ser colocadas "sobre" a solução ArcSight por forma a ampliar a sua capacidade em certos aspectos.

O ArcSight NCM/TRM (*Network Configuration Manager* e *Threat Response Manager*) é um desses complementos. Trata-se de um dispositivo que, alavancando o conhecimento detido sobre a infra-estrutura com a qual lida (topologia de rede ou configurações de equipamentos de rede como *switches* ou *firewalls*) e ferramentas de gestão centralizada de recursos e configurações de rede, permite facilitar e melhorar a rapidez na resposta a incidentes de segurança, no que toca a desencadear acções correctivas ou preventivas, para mitigar os efeitos de ameaças. Pode agir, por exemplo, isolando nós de uma rede ou aplicando políticas e regras de *firewall* de forma a bloquear ligações de determinados endereços IP.

O ArcSight *ArcMC* (*Management Center*) é outra ferramenta complementar, concebida com o propósito de consolidar actividades administrativas dos principais equipamentos/componentes (*Logger e Connector Appliance*) através de uma interface única.

Para além destas 2 alternativas, existem numerosos pacotes de conteúdo, distribuídos separadamente, ou incluídos em conjuntos da solução, que podem ser adicionados à plataforma, provendo conteúdo para, por exemplo: aferir e validar aspectos de conformidade (com normas como a HIPAA, SOX ou outras referências da indústria), detectar padrões e construir regras e alertas com base nos mesmos, e até automatizar o processo de análise de risco [8].

2.2.3 Recursos de Conteúdo

Estando os componentes de *hardware* e *software necessários* a postos, podem então passar, quer sejam os utilizadores, administradores ou analistas, a criar e usufruir dos conteúdos pré-existentes, através das interfaces já mencionadas. Este conteúdo pode apresentar-se em diversos formatos e é utilizado para analisar dados previamente recolhidos.

Os **recursos** são os constituintes fundamentais daquilo que formará a informação utilizável por todos utilizadores do SIEM, dentro da organização. Para além das características genéricas como **nome** e **identificadores internos** à plataforma, cada um possui especificidades adequadas à sua função. Apresentam-se, em seguida, alguns desses recursos, de forma não exaustiva [10]:

- ***Active Channels***, ou ***canais activos*** – instrumentos de visualização para apresentar conjuntos de eventos que podem estar a ser recolhidos em tempo-real ou que foram recolhidos durante uma janela temporal definida. Os dados apresentados podem, também, ser refinados através de filtros ou regras. A figura 4 revela um canal activo que apresenta resultados com base num filtro definido ("eventos de correlação e provenientes de uma determinada fonte de dados") bem como várias estatísticas sobre os eventos seleccionados. A informação pode ser apresentada com um conjunto de campos (associados ao evento em si), definidos pelo utilizador ou utilizando uma configuração por omissão da plataforma. É ainda possível obter mais informação sobre os eventos/sequências de eventos, em específico, utilizando as capacidades de análise da plataforma (por exemplo, o evento pode ser seleccionado com o rato e será disponibilizada informação mais específica).

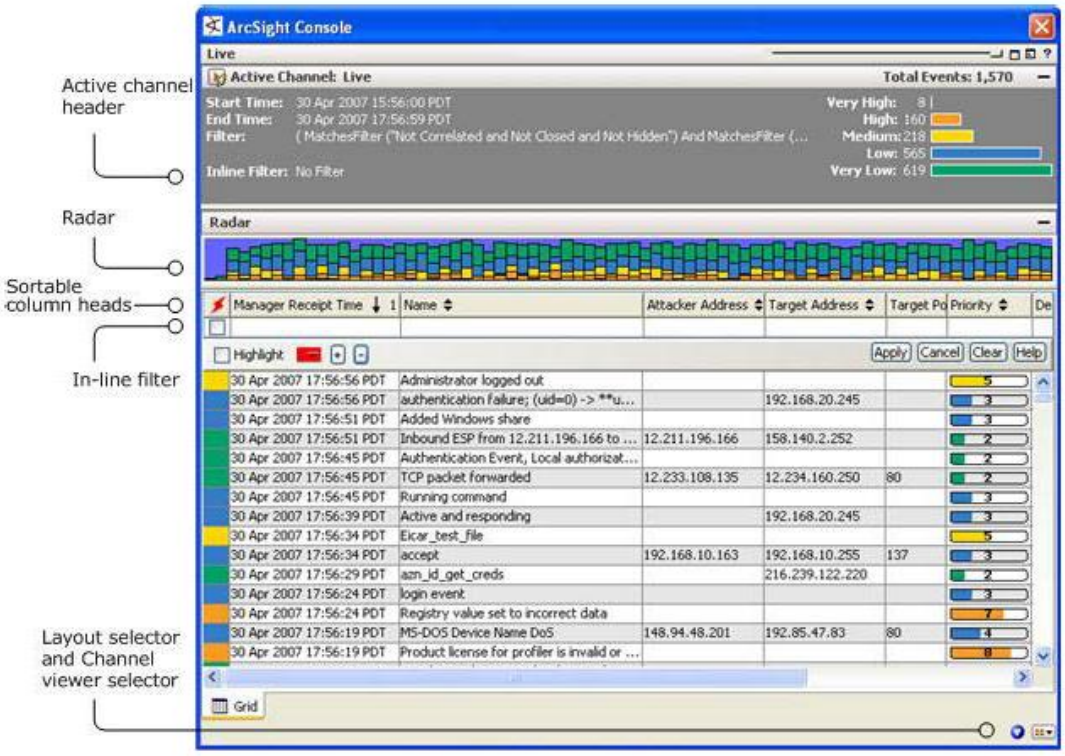

Figura 4: Active Channels [9]

- ***Regras*** – os recursos de utilização mais evidente, facilitados pelo motor de correlação do sistema. Estes recursos têm a capacidade de desencadear acções ou emitir alertas aquando da observação de um conjunto de ocorrências no sistema. Elas podem correlacionar eventos e procurar sequências ou aglomerados de eventos durante um período de tempo definido. A figura 5 demonstra o método de criação de regras e,

como podemos ver pelas abas superiores, para cada regra, podem ser definidas características como: condições de disparo (formato idêntico ao de uma *query*), condições de agregação dos eventos (aglomeração com base num certo campo, por exemplo) e acções a desencadear em caso de disparo (podem ser, desde a lançar um alerta para uma equipa de resposta a incidentes, até desencadear a execução de um *script* interno).

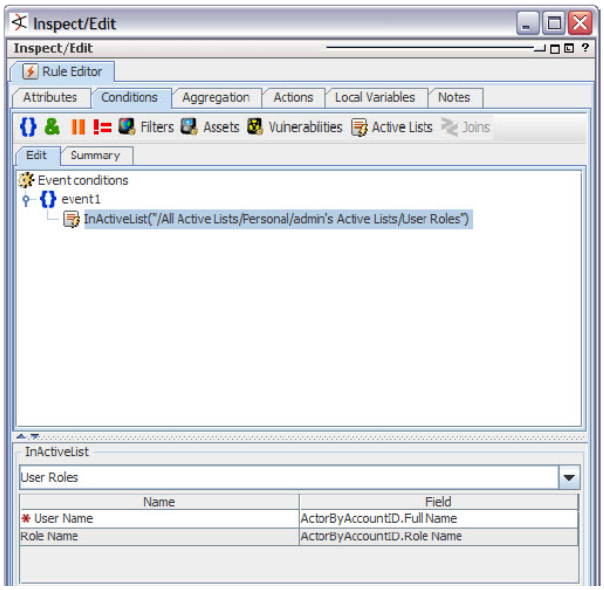

Figura 5: Rules Creation Wizard [9]

- ***Filtros*** – os recursos mais simples do ambiente que, como o nome sugere, são utilizados para filtrar o *input*/*output* de outros recursos mais complexos, com base num conjunto de condições. A figura 6 mostra o utilitário de configuração de filtros, no qual se indicam as condições do mesmo (em formato de condições de uma *query*).

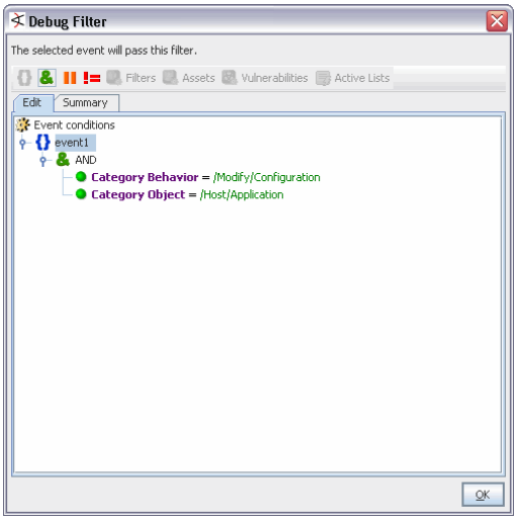

Figura 6: Filter Creation Wizard [9]

- ***Data Monitors,*** ou ***monitores de dados*** – recursos focados na apresentação de dados em tempo-real, segundo um formato específico (tabelas, gráficos, etc.) e de acordo com um conjunto de condições (um filtro, por norma). São utilizados em painéis de visualização para agregar conjuntos de elementos de visualização semelhantes. Podem apresentar várias estatísticas expondo, geralmente, uma perspectiva cronológica sobre os dados. Têm também a capacidade de apresentar os dados de forma mais simplista através de, por exemplo, tabelas de eventos.

- ***Dashboards*** ou ***painéis de visualização*** – janelas de apresentação simples, utilizadas para visualizar conjuntos de monitores de dados numa única posição (conjunto de tabelas, por exemplo). A figura 7 mostra um painel de visualização composto de 5 monitores de dados distintos. Como podemos ver, os dados/eventos podem ser apresentados sob a forma de tabelas, gráficos com contagens (apresentado na imagem, um gráfico com os protocolos de transporte mais usados e referências de cor para a severidade dos eventos recolhidos) e outros mais apelativos, como gráficos sectoriais (vejam-se os gráficos com zonas mais acedidas ou percentagens do total de eventos, agrupadas por IP de origem) ou de barras (caso do gráfico que apresenta o topo de categorias dos eventos registados pelo sistema).

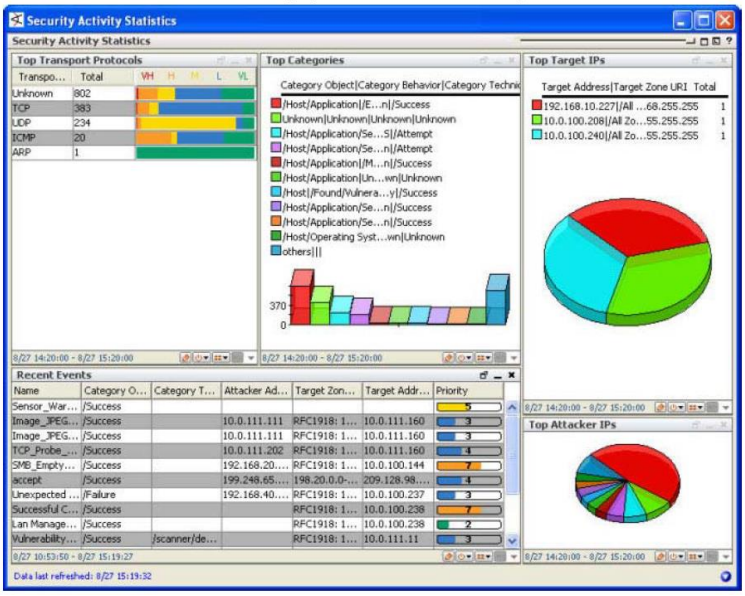

Figura 7: Data Monitors in Dashboards [9]

- ***Queries*** ou ***pesquisas*** – operam sobre eventos do passado, captando dados de acordo com um filtro e seleccionando apenas um conjunto específico de campos do

modelo de dados CEF, para cada evento. É também possível configurarem-se alguns aspectos de agregação desses eventos, reduzindo, assim, o volume de dados seleccionado para análise. A figura 8 apresenta o utilitário de criação de pesquisas que podem ser definidas segundo parâmetros como o instante de começo e de fim da busca (indicam o período sobre o qual ela vai incidir, quando a pesquisa é despoletada), o número máximo de linhas/eventos a recolher e os campos a recolher dos eventos que estejam conforme as condições definidas no filtro.

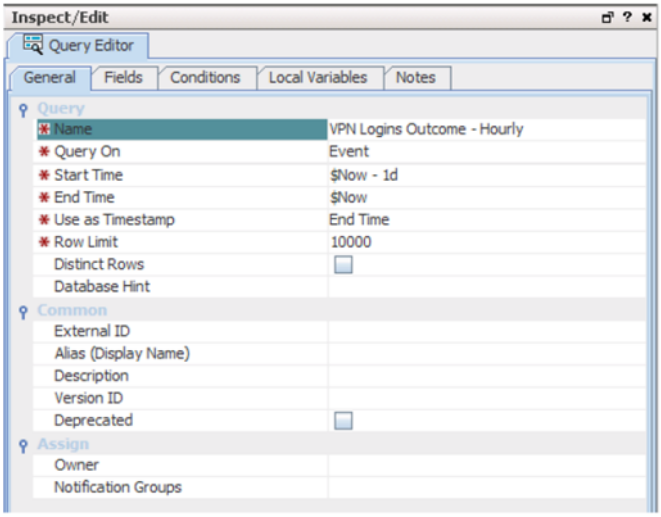

Figura 8: Query Creation Wizard [9]

- ***Trends*** ou ***tabelas de recolha*** – são tabelas de armazenamento de eventos, com grandes dimensões que, periodicamente (conforme configurado pelo utilizador) recebem dados recolhidos com base nas características de uma determinada **pesquisa**. A figura 9 demonstra o utilitário de criação dessas tabelas e a configuração de parâmetros de agendamento e de retenção dos dados (são utilizadas como tarefas periódicas e o seu conteúdo é actualizado consoante o agendamento configurado).

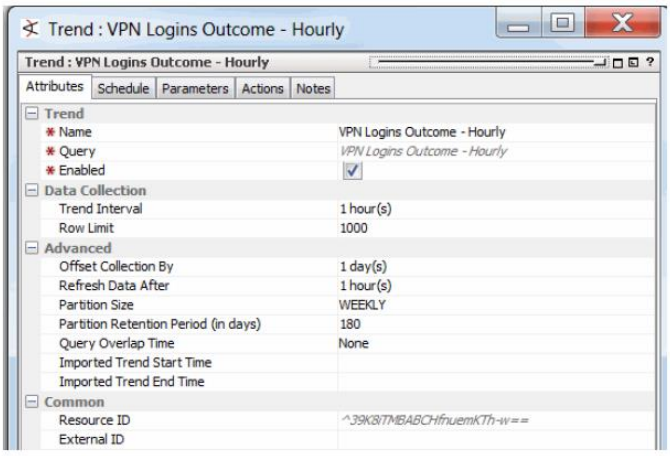

Figura 9: Trend Creation Wizard [9]

- ***Active Lists*** ou ***listas activas*** – são listas regulares que vão sendo preenchidas com dados de eventos, podendo ser reutilizadas para refinar ou modificar outros recursos (como por exemplo, uma regra que procure por *logins* de um determinado utilizador que foi colocado numa *Active List* contendo identificações de utilizadores não confiáveis). Estas listas podem também ser preenchidas de forma automática através de acções desencadeadas por regras. A figura 10 apresenta uma lista activa configurada para reter informações sobre endereços IP identificados como maliciosos, como o próprio IP e referências da zona onde o IP se encontra.

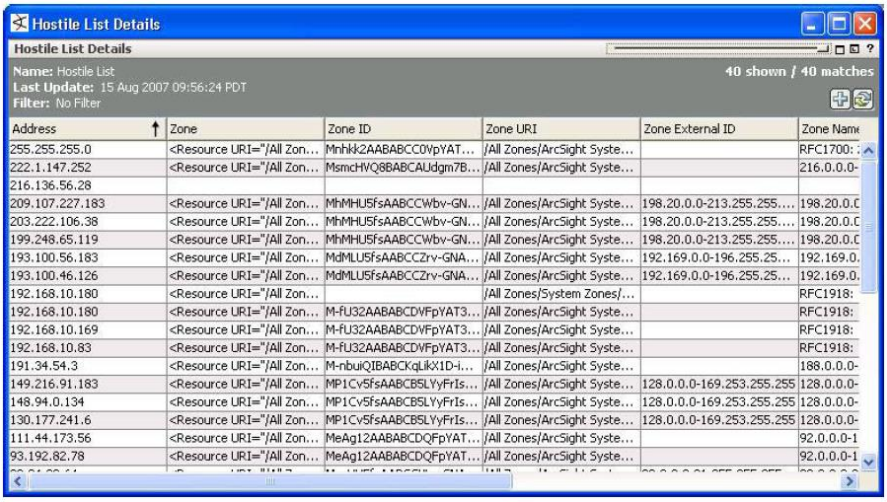

Figura 10: Exemplo de Active List [9]

- ***Integration Commands*** ou ***comandos de integração*** – são pequenas ferramentas configuráveis, para facilitar a interoperabilidade com outros sistemas, deixando margem para o utilizador poder desencadear acções de forma directa, a partir do próprio ambiente da plataforma. Utilizam-se principalmente para executar, pontualmente, scripts ou realizar pesquisas, com informações contidas em eventos recolhidos, em sites *web* como o Virus Total, por exemplo. A figura 11 mostra alguns exemplos de comandos de integração que se encontram instalados, por omissão, na plataforma e que têm funcionalidade como, por exemplo, pesquisar no Google pelo conteúdo do campo seleccionado ou despoletar comandos como um *nslookup* ou um *traceroute*. Este tipo de recursos pode, em regra, ser despoletado em qualquer ambiente de consola em que se encontrem listados eventos, através do botão direito do rato.

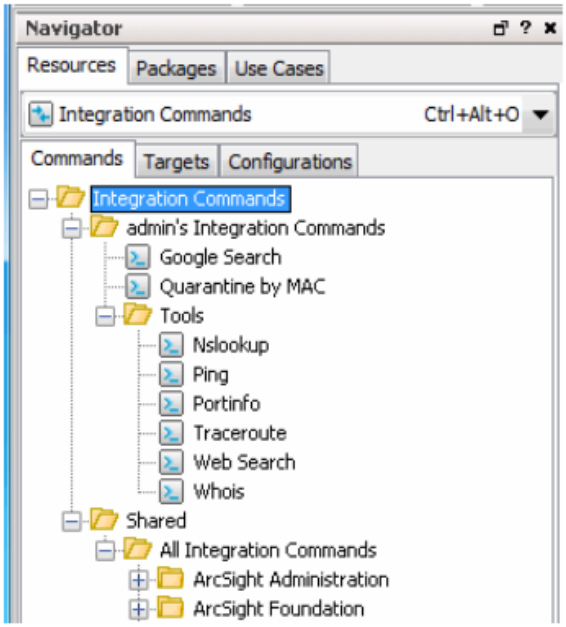

Figura 11: Integration Commands [9]

Existem muitos outros recursos que acrescentam à funcionalidade base, já fornecida pelos atrás mencionados, aumentando as capacidades de analítica e interacção da plataforma. Os listados, no entanto, e como pudemos observar ao longo do projecto, compõem um núcleo essencial, usado largamente por centros CSOC nas actividades diárias de controlo e resposta a incidentes. Finalmente, é deste subconjunto que retiramos algum conteúdo para auxiliar à demonstração da nossa prova de conceito.

2.3 *Open-Source Intelligence* (OSINT)

Nesta secção pretendemos dar uma visão geral tanto do conceito, como do contexto histórico em que o mesmo surge (OSINT, ou informação estratégica de fontes abertas). Não é nosso objectivo sermos exaustivos na explanação de toda a envolvência histórica e de todas as subtilezas que nos trouxeram ao estado actual do sector, mas sim o de disponibilizar uma base para que se compreenda a necessidade de construir uma ponte entre o assunto em questão (OSINT) e o paradigma fundamental desta tese (SIEM).

A *Open-Source Intelligence* (OSINT ou OSCINT), enquanto conceito, foi explicitamente definida e caracterizada pela organização Open Source Solutions, Inc. no ano de 1997. Numa publicação, a mesma explicava-o como sendo conhecimento estratégico, disponível publicamente e passível de livre utilização, circulante em meios de comunicação tais como a televisão, a rádio ou os *media* impressos. É dada particular relevância ao facto de que se prevê que essa mesma informação não contenha qualquer tipo de dados confidenciais, sensíveis ou ao abrigo de direitos de autor/especiais que os

proíbam de tal recolha. Adicionalmente, salientava-se que essa mesma recolha não poderia ser realizada de forma "clandestina" ou "encoberta" [10].

Esta definição do conceito é de grande importância, tanto para o nosso trabalho, como para a compreensão geral do assunto. Ela coloca a OSINT numa perspectiva disjunta da dos canais encobertos ou subversivos através dos quais são, hoje em dia, ilegal ou deslealmente, obtidas grandes quantidades de informação. Podemos comprová-lo observando práticas menos próprias de governos, organizações e até instituições sancionadas e suportadas por nações-estado, que optamos por não mencionar. Este paradigma está visivelmente afastado dessas abordagens, já que a informação que flui nos seus canais é de fácil acesso e pode ser utilizada por qualquer individuo, ainda que, algumas vezes, com restrições (em tudo semelhante à liberdade de expressão) [11, 12, 10]. De certa forma, acaba por se levantar um paralelismo entre a liberdade de "recolha" e a liberdade de expressão, sendo ainda mais perceptível como o segundo se pode entender como sendo a contraparte do primeiro.

A OSINT é, e tem sido, amplamente utilizada por instituições governamentais como a (Norte Americana) *Federal Research Division*, que desempenha actividades de investigação e análise nestas fontes abertas (OSINTs), para várias outras agências, ou a (também Norte Americana) *Defense Intelligence Agency*, que conduz operações de espionagem militar além-fronteiras. Historicamente, o uso dessa mesma via, e dos seus recursos, arrecadou grandes quantidades de informação estratégica de grande valor para as operações que dela usufruíram [13].

Informação estratégica de fontes abertas já era de uso corrente há bastante tempo, sendo difícil separá-la de actividades como a recolha secreta de informação, mas apenas atingiu um nível de maturidade razoável com o intensificar do conflito durante a 2ª Guerra Mundial, quando o seu potencial foi levado ao máximo, através do usufruto das suas contribuições em contexto bélico.

A criação, nos Estados Unidos, do FBMS (*Federal Broadcast Monitoring Service*, que mais tarde formou o *Federal Intelligence Monitoring Service*, sob a alçada da CIA), para o propósito de monitorizar as comunicações de rádio de onda curta durante a guerra, traduzindo e analisando a informação circulante, é fulcral para este processo (de atribuir relevância ao paradigma) e, visto por muitos, como sendo um marco que desencadeou uma grande evolução na área. Seguiram-se os Britânicos, ordenando, em 1939, a que a emissora BBC escrutinasse informação estrangeira (essencialmente dos meios impressos radiofónicos). Durante os períodos que sucederam a Grande Guerra e, especialmente, com o clima de elevada tensão trazido pela Guerra Fria, surgiram várias iniciativas "clandestinas", ou encobertas, de recolha de informação estratégica, erguidas tanto por potências Ocidentais como Orientais [13, 14].

Com o aumento da interligação de comunicações e da facilidade no acesso aos meios de difusão de informação (como os *media* e a internet) por todo o mundo, a área da informação estratégica acabou por ser também arrastada na corrente, tornando-se mais e mais dispersa e acessível. O número de canais através dos quais a informação podia ser transmitida aumentou exponencialmente (com a introdução das plataformas de comunicação *web* como fóruns, *message boards*, *mailing lists*, etc.) acabando as fontes abertas por se demonstrarem, gradualmente, como sendo alternativas muito cativantes às abordagens tradicionais e encobertas de recolha de informação [3].

Construía-se, a passo e passo, um vasto universo de informação estratégica gratuita, rápida e facilmente acessível, por todo o mundo.

O interesse bem definido e direccionado, por parte do sector privado (não considerado parcerias cunhadas entre os sectores público e privado), no entanto, surge apenas mais tarde. Deve-se isso ao facto dos benefícios que a OSINT poderia oferecer a este sector só começar a ser compreendido com o crescente interesse que as organizações revelavam face a áreas como *Business Intelligence*, *Commercial Intelligence* e *Competitor Intelligence*. A segurança informática (da perspectiva do analista de segurança) e a "gestão de relação com o cliente" (ou CRM, *Customer Relationship Management*) são duas áreas mais específicas que também acabaram por colher parte dos benefícios de fontes derivadas da OSINT [10, 15].

Especificamente no âmbito da segurança informática, este paradigma tem recebido (e continua a receber) bastante relevância pelo valor que tem a sua capacidade de auxiliar na predição, prevenção e detecção de ataques (e ameaças) dirigidos a uma organização. Apesar da sua importância, os processos da sua recolha, análise e interligação com outros métodos e ferramentas de suporte analítico, é ainda muito custoso e exigente. Por norma, essa tarefa é desempenhada manualmente por analistas associados a algum Centro de Resposta a Incidentes/Operações de (Ciber-)Segurança. Embora surjam alguns esforços (essencialmente focados na porção referente à recolha de informação, como é o exemplo do Maltego ou do FOCA) no sentido de facilitar essas tarefas, parte substancial do trabalho continua a ser fundamentalmente manual, servindo-se pouco de automatismos ou técnicas de aprendizagem de máquinas [16, 17, 18].

2.4 Proposta

A figura 12 esquematiza a proposta da arquitectura da ferramenta, desenvolvida para colmatar o problema em causa, de transportar informação, refinada, de fluxos OSINT para plataformas SIEM, de forma automatizada.

Ao idealizarmos (e mais tarde, ao concebermos) a solução, mantivemos o objectivo de criar uma estrutura hierárquica e modular. Acreditamos que isto nos permite abrir a ferramenta a melhorias futuras e permitir a validação isolada do funcionamento correcto de cada um dos subcomponentes.

No que toca à ideia de versatilidade, ou extensibilidade, parece-nos que a opção de desenho permite uma integração mais facilitada, no futuro, e consoante a necessidade, de novas fontes de dados (novos fóruns, novas APIs, novas redes sociais, etc.), novas plataformas SIEM (apesar do nosso foco ser sobre o ArcSight, compreendemos a necessidade de interoperabilidade dos produtos) e novos mecanismos que aumentem as capacidades do nosso componente central, o "**motor**".

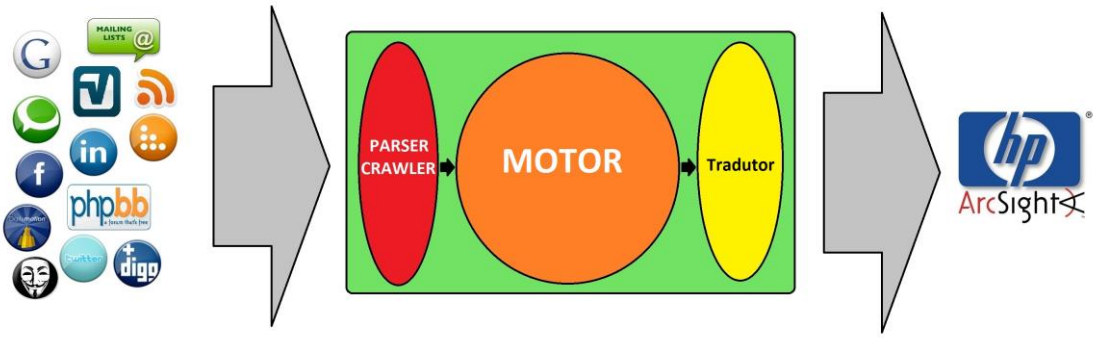

Figura 12: Arquitectura da ferramenta proposta

Como a imagem anterior evidencia, dividimos a ferramenta em 3 componentes (ou grupos de componentes) principais que, de uma forma geral: captam a informação de fontes abertas configuradas pelo utilizador (como demonstrado mais à esquerda na imagem), processam e reduzem essa informação, e passam, depois, a alimentar o ArcSight (ou outra plataforma SIEM configurada pelo utilizador) com a mesma.

O primeiro componente, intitulado "***parser crawler***", é, como o nome indica, um *crawler* que analisa e normaliza os dados que vai tratando, através de operações de "*parsing*"[2].

De acordo com esta definição, o nosso "***parser crawler***" está, então, incumbido das tarefas de procurar e analisar informação das fontes para as quais é apontado, utilizando quaisquer meios necessários (e que estejam à sua disposição) para obter a dita informação, quer seja através de APIs ou de *downloads* de páginas na íntegra, por exemplo. Os dados recolhidos através deste processo são posteriormente entregues ao componente seguinte, o "**motor**".

[2] Tomamos a palavra "*crawler*" como sendo referente a um agente automatizado de recolha de informação para a qual é direccionado; e a palavra "*parser*" como caracterizando um agente semelhante que, ao invés de recolher, opera sobre a informação já colectada, analisando-a no sentido de revelar, dentro da mesma, os dados que lhe são requisitados.

É neste nosso "**motor**" que está contida a maior parte da estratégia da nossa solução, já que foi através dela que tentámos simular as capacidades analíticas dos analistas de SOC que usufruem, regularmente e de forma manual, dos benefícios da informação OSINT. É sobre este componente que recai a tarefa de conduzir uma análise mais específica com vista a reduzir o volume de informação a um mínimo que é tão **completo** (contém todos os resultados relevantes) quanto **preciso** (contém apenas a informação essencial, eliminando o máximo possível de irrelevância nos dados), com base nos critérios de relevância definidos (como veremos mais adiante).

O conteúdo produzido pelo "**motor**" é, seguidamente, conduzido para o "**tradutor**" (***wrapper***) para que os dados sejam normalizados e mapeados de acordo com o modelo de dados da plataforma SIEM em causa e, então, encaminhados para a mesma.

Um subcomponente não evidente na representação (apresentado em detalhe na secção 3.6) mas que possui também um papel fundamental no comportamento da solução é o instrumento responsável pela passagem do *feedback* do utilizador ao **motor**. Trata-se de um utilitário de que o utilizador se servirá para indicar à plataforma aquilo que ele entende como sendo conteúdo relevante, por forma a condicionar análises de informação futuras, aumentando a sua **precisão** (na expectativa de refinar os resultados produzidos).

A nossa prova de conceito (presente na secção 4) será sustentada pela plataforma ArcSight e servir-se-á de informação presente na rede social Twitter. Como método de avaliação começamos por sleccionar as palavras-chave que guiam a recolha na rede social em causa; prosseguimos a definir a condição de relevância da informação (para podermos caracterizar o conteúdo debitado pela nossa solução na plataforma SIEM); passamos, seguidamente, à recolha de informação com e sem mecanismo de refinamento; finalmente, comparamos os resultados obtidos e discutimo-los.

É com este protótipo que conseguimos, através de evidência empírica, afirmar que a nossa solução contribui positivamente para o seu propósito, constituindo uma abordagem de aplicabilidade viável e comprovada.

2.5 Trabalho Relacionado

A área de *data mining* (extracção de dados) é próspera em trabalhos que incidem sobre técnicas de extracção de informação de redes sociais, tanto pela riqueza como abundância e fácil acesso à informação presente nas mesmas. Para além de trabalhos percursores e mais genéricos, com o intuito de constituir as bases teóricas da abordagem e definir problemas, limitações e aplicações práticas [19, 20, 21], existem outros, mais direccionados aos nossos objectivos.

Rahman, em 2012, procura extrair conhecimento de contas da rede social Facebook e inferir informação mais complexa a partir de dados e características simples do indivíduo, disponíveis no seu perfil [22]. Amine et al. desenvolve abordagens mais complexas para detectar opiniões/sentimento em *tweets* [23] enquanto Zhu procura derivar autenticidade em opiniões utilizando, adicionalmente, um mecanismo de votação, num esforço de aprumar os resultados da análise [24].

Outros trabalhos mais ambiciosos visam servir-se das redes sociais, tentando retirar proveito do seu cariz preditivo, isto é, tentando abordá-las para extrair informação de uso em modelação preditiva para, por exemplo, a previsão de resultados de jogos de futebol [25] ou do sucesso de campanhas de marketing [26]. Há ainda outros que alavancam técnicas de aprendizagem e extracção supervisionada [27] utilizando a informação disponibilizada pelos próprios utilizadores dentro da rede [28, 29].

No mercado encontramos esta vertente de extracção de dados, em alguns casos para análise sentimental, mais focada na área do CRM, ou gestão de relação com o cliente. Como exemplo podemos citar os produtos comerciais Nimble [30], Salesforce [31] e Zoho [32]. Aqui a abordagem comum foca-se em detectar interacções por parte do cliente que possam ter impacto sobre o negócio e alguns incluem capacidades de resposta automática, com recurso a inteligência artificial [33].

Incidindo mais sobre o tema da segurança, outros trabalhos surgem, focados na obtenção de informação estratégica, e de segurança, a partir de redes sociais. Jurdak et al. [34] e Matias et al. [35] tentam, em trabalhos menos relacionados com o nosso tema, analisar e automatizar a detecção de ameaças à biossegurança e de outros problemas como o assédio em redes sociais. Já noutros trabalhos como Gharibi e Shaabi [36], Guille e Favre [37] e Miller et al [38] estas capacidades analíticas e de aprendizagem e extracção de informação são transportadas para a detecção de ameaças associadas ao espectro da cibersegurança.

Começamos a ver, assim, esforços iniciais de conjugação de técnicas de inteligência artificial associada à cibersegurança, como é o caso de Dilek et al. [39] que procura esquematizar avanços realizados até à data sobre aplicações dentro do tema mencionado. Zope et al. [40] propõe uma abordagem mais especializada, manipulando técnicas de extracção de informação e de visualização, no âmbito do SIEM.

Quanto a avanços e esforços de investigação na área da OSINT, ou informação estratégica de fontes abertas, observa-se uma tendência similar de transporte e adopção de técnicas de extracção de informação em meios regulares como a internet, e gradual orientação/foco sobre a informação circulante nas redes sociais. Maciolek e Dobrowolski [41] propõem uma *framework* adequada à extracção de informação de grandes

quantidades de dados, como as encontradas em fluxos OSINT na internet. De Santos e Vega [42] tentam teorizar vários aspectos da OSINT.

A introdução destes esforços e técnicas na área da cibersegurança torna-se, então, um passo lógico e podemos observar trabalho significativo no sentido de advogar o uso de informação dos fluxos já mencionados para enriquecer e fortalecer o combate e prevenção de ameaças informáticas [43, 16, 44]. Também no ramo comercial (do mercado) vemos produtos que procuram resultados semelhantes, como o McAfee ZeroFox Attack Detection System [45] (complemento aos restantes produtos de segurança do mesmo fabricante), ThreatConnect [46], Cyveillance [47] e Soteria [48].

Estes produtos aproximam-se do nosso âmbito, no entanto, o seu foco não é o de direccionar a informação que produzem ao SIEM mas sim o de disponibilizar ao utilizador *threat intelligence* associada às redes sociais.

Especificando um pouco mais, alguns projectos surgiram nos últimos anos, incidindo sobre a necessidade de transportar esse tipo de informação já mencionado para as plataformas SIEM. Projectos "open source" como o ArcOSI [49] e o Collective Intelligence Framework [50] tiveram um papel percursor nesta iniciativa, procurando, de forma rudimentar, transportar informação já tratada e analisada sobre, por exemplo, listagens domínios com reputação de serem maliciosos, para o SIEM. Alguns produtos existem também com o mesmo intuito, como por exemplo, o ArcReactor [51] e o serviço prestado pela ThreatStream [52].

Neste caso, os produtos mencionados focam-se no nosso âmbito em específico. Apesar das capacidades serem semelhantes e de os mesmos procurarem introduzir informação estratégica no SIEM, qualquer das ferramentas tenta fazê-lo recorrendo a informação presente na internet em *sites* de reputação de domínios ou IPs (informação já detectada como relevante). A excepção é o ThreatStream, cujo produto é um serviço de subscrição em que um fluxo de informação é criado para colocação no SIEM – a informação quanto ao produto é pouco clara mas pela escassez de interacção parece indicar que esse fluxo não é configurável. [52]

Das alternativas acima mencionadas, vemos algumas que se aproximam do trabalho que aqui propomos, como é o caso do ArcOSI ou do ArcReactor que extraem listagens de domínios e endereços IP maliciosos, de sites que actualizam constantemente as suas bases de dados, ou o Collective Intelligence Framework que tenta reunir conhecimentos e descobertas de várias equipas que usem a plataforma e consolidar e distribuir essa informação pela sua rede. A ThreatStream oferece serviços de envio de informação estratégica de segurança para as empresas que os subscrevam, podendo esta informação tratar-se de dados para integração no SIEM.

Apesar das semelhanças, os objectivos são algo díspares dos nossos (o foco nem sempre é o SIEM, as redes sociais ou o refinamento automática da informação) e não conseguimos encontrar trabalhos ou produtos que tentassem tratar este tema directamente, de recolher informação de fluxos de OSINT, processá-los através de técnicas de aprendizagem de máquinas e extracção de conhecimento para reduzir o volume de informação e filtrar a informação irrelevante, e passa-la a plataformas SIEM através de componentes de integração.

Nesta secção procurámos dar algum enquadramento sobre os temas que envolvem o trabalho em causa. Começámos por contextualizar o trabalho em termos dos sistemas SIEM (o que são, a sua utilidade e os fundamentos básicos do seu funcionamento), tendo, em seguida, explicitado alguma informação sobre um sistema em concreto, denominado ArcSight. Após darmos algum conhecimento sobre os principais constituintes do mesmo e sobre a sua arquitectura, passámos a definir a OSINT, a sua relevância (não só em referência a este trabalho, mas também no seu contexto histórico) e a necessidade de a transportar para o paradigma do SIEM (e da segurança informática, em geral). Passámos, então, a explanar a arquitectura e o racional por detrás da solução que propomos para o problema em estudo. Finalmente, descrevemos alguns trabalhos relacionados com o nosso e esforços que contribuem para o tema em estudo (e temas da periferia).

Na secção seguinte apresentaremos em detalhe a solução que já identificámos e todos os seus componentes.

Capítulo 3

Solução

Nesta secção iremos explicitar os detalhes do funcionamento e desenho do protótipo que desenvolvemos para integrar informação de fontes abertas no SIEM. Para além de aspectos como a arquitectura (ou a maneira como os componentes interagem), incluímos detalhe sobre a maneira como cada componente foi construído e é utilizado.

A figura 13 esquematiza a arquitectura da ferramenta desenvolvida neste trabalho e utilizada na nossa prova de conceito. Apesar de funcionar como um componente único, é constituída por vários *scripts* e módulos que, apesar de poderem funcionar isoladamente, formam uma única plataforma.

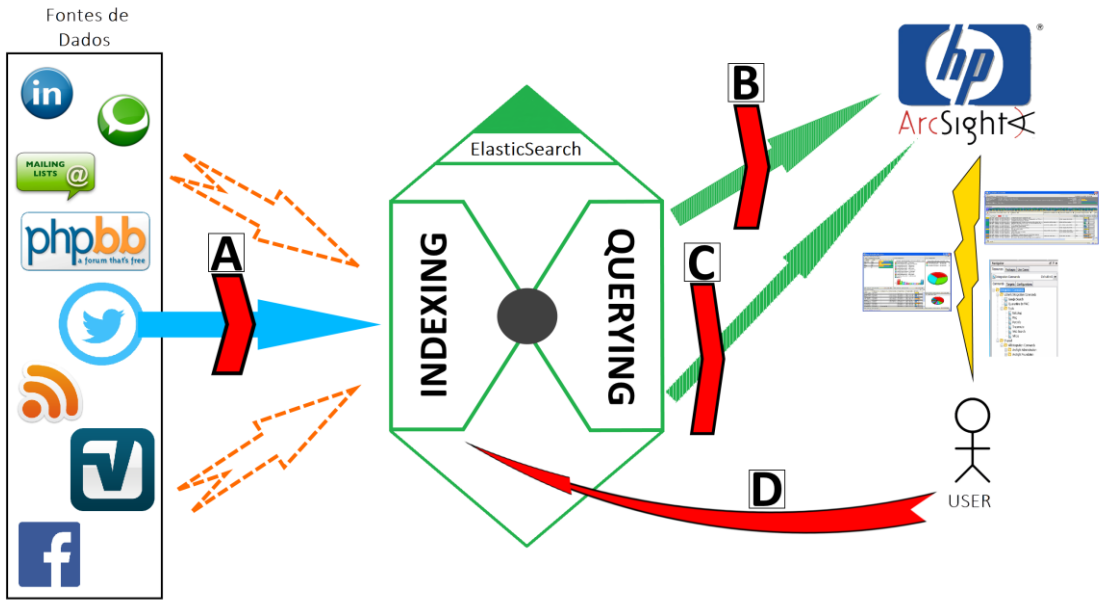

Figura 13: Arquitectura e funcionamento da ferramenta desenvolvida

A ferramenta foi colocada e testada no ambiente de testes da Unisys, juntamente com um equipamento HP ArcSight *Express* e as análises sustentadas pela plataforma, e seu conteúdo, foram validadas por colaboradores experientes da equipa de segurança da empresa.

De um modo geral convém especificar o seguinte: os *scripts* e módulos são representados pelas setas vermelhas e com contorno preto e são referenciados conforme a letra que lhes corresponde, e qualquer seta no esquema é representativa de um fluxo de informação, tratado ou não, sendo que as setas com contorno tracejado indicam possibilidades não materializadas no nosso protótipo.

As secções apresentadas de seguida explicitam o funcionamento da ferramenta e detalham, também, alguns aspectos da implementação que poderão não ser perceptíveis através do esquema.

3.1 Arquitectura

Podemos observar, através da figura 13 que se mantém a estrutura idealizada inicialmente, que pode ser repartida nos 3 principais planos lógicos da solução, a saber:

- O "***parser crawler***" - aqui representado enquanto plano (já que vai abarcar todos os eventuais módulos que integrem novas fontes de dados na solução) é composto pelo *script* apresentado à esquerda (**A**) que extrai informação da rede social Twitter e a encaminha para o ElasticSearch [53] (o nosso **motor**)
- O **motor** - componente nuclear da solução, materializa-se através do *software open-source* ElasticSearch [53] que é composto, de uma forma abstracta, por dois módulos: de indexação ("**INDEXING**") e extracção ("**QUERYING**") de informação.
- Por fim, o **tradutor** (***wrapper***) apresenta-se sob a forma de um conjunto de *scripts* (**B** e **C**) que são responsáveis pela formatação e envio da informação para o ArcSight (no caso da nossa prova de conceito).

Um outro componente que não inserimos explicitamente em qualquer dos planos, pela sua versatilidade e por se poder situar, facilmente, em qualquer dos planos, é o *script* (**D**) que faz a ponte entre o utilizador e a plataforma, inserindo informação de *feedback* dada pelo utilizador no ElasticSearch.

3.2 *Parsing*/*Crawling* de informação

Na nossa ferramenta, tendo em conta a prova de conceito, integramos dados da rede social Twitter (mais concretamente, os *tweets* publicados pelos utilizadores) no ElasticSearch através de um *script* em Python designado "tweetsniff.py" (assinalado na figura 13 com a letra **A**), cujo código se encontra no anexo A.

Este pequeno aplicativo, baseado no projecto de código aberto de Xavier Mertens [54] serve-se de uma API, criada na mesma linguagem de programação, para interagir com a interface REST da plataforma de desenvolvedores do Twitter e recolher dados com

base num conjunto de configurações fornecidas, como as palavras-chave e contas a serem monitorizadas, e a localização do servidor em que está a correr a instância do ElasticSearch.

Este componente não possui, por si só, a capacidade de filtrar os resultados, para além de os restringir através da própria plataforma do Twitter, apenas aos que contenham determinadas palavras-chave. Serve-se, no entanto, da funcionalidade de "*search*", ou procura/busca, da plataforma. Esta funcionalidade permite a qualquer utilizador, em qualquer parte do mundo, observar *tweets* que contenham uma determinada palavra-chave, já que qualquer publicação de qualquer membro da rede está visível a qualquer outro membro. Adicionalmente, é também capaz de utilizar a própria *timeline*[3] da conta "de serviço", não fazendo, neste segundo caso, filtragem.

A execução do mesmo decorre em 3 principais passos:

1. Em primeiro lugar, é iniciada uma ligação à API, com os dados da conta de que nos servimos para criar este acesso, através da interface open source "Python-Twitter". São também analisadas as informações de configuração passadas através de um ficheiro indicado pela da linha de comandos. Este ficheiro deverá conter: os dados de ligação ao servidor do ElasticSearch, os dados secretos da conta do Twitter utilizada e as palavras-chave sobre as quais vai incidir a pesquisa (um exemplo encontra-se no anexo E).
2. Seguidamente, a aplicação vai, com uma periodicidade definida automaticamente e calculada através dum método interno à API (que, consoante o volume de dados que estão a ser retirados, computa um período de espera de forma a não exceder as limitações de pedidos, impostas pelo Twitter[4]), actualizar os conteúdos da *timeline* e realizar uma pesquisa sobre cada palavra-chave indicada no ficheiro de configuração.

[3] A *timeline* do Twitter é um registo de cada conta que contém um sumário das actualizações de estado submetidas pelos utilizadores que a mesma está a "seguir", isto é, agindo como um subscritor de eventos.

[4] Para o efeito da nossa ferramenta, convém considerar dois limites de utilização da API, impostos pelo Twitter: os pedidos referentes a "procuras" estão limitados a 450 por cada 15 minutos (cada pedido corresponde a uma página e pode devolver até 100 *tweets*) e os pedidos de actualização da *timeline*, a 300 pela mesma unidade de tempo (cada pedido corresponde a uma página que pode conter até um máximo de 200 *tweets*). Adicionalmente convém referir que cada procura pode capturar informação com data de publicação máxima até 6-9 dias, ou 1500 *tweets* (o que primeiro se verificar) e que as actualizações da *timeline* podem extrair até um máximo de 800 *tweets*.

3. Por último, é desencadeado o processo de indexação de dados na ferramenta ElasticSearch. Os *tweets* extraídos são colocados, em formato JSON, dentro da plataforma e sob a forma de documentos compostos pelos seguintes campos: publicação efectuada (texto publicado), utilizador que efectuou e outros conteúdos mais específicos, como identificadores internos e referências para outras publicações associadas. Findada esta porção da execução, o ciclo completa-se com o "adormecimento" do processo pela duração dada pela API (como mencionado no ponto 2), sendo reexecutados o segundo e terceiro passos por tempo indefinido.

A explicação anterior apresenta apenas os aspectos mais "alto-nível" do funcionamento da solução. Para que as actividades básicas sejam cumpridas e a recolha seja bem-sucedida, é, no entanto, necessário que tenhamos definidos alguns mecanismos adicionais, a saber:

- Cada interacção com a API devolve (para além dos conteúdos óbvios como os *tweets* resultantes da procura) um "*search_id*" que é, basicamente, um identificador do último resultado presente no conjunto devolvido. Para manter a coerência dos dados que vão ser indexados, este identificador é guardado no disco, após os *tweets* serem colocados no ElasticSearch. Apesar de este método conservar a visão sobre os dados que vão sendo extraídos através de pedidos sucessivos e sequenciais, em caso da execução do *script* parar antes da gravação, em disco, do *search_id*, podem perder-se alguns dados ou repetir-se a indexação de outros, gerando duplicados no sistema.
- Para lidar com o problema dos duplicados, servimo-nos das capacidade internas do ElasticSearch que, aquando da indexação de conteúdos foi configurado para detectar conteúdo duplicado e eliminá-lo [53].

3.3 ElasticSearch

O ElasticSearch [54] (visível na figura 13 como o hexágono central) é um projecto *open-source* desenvolvido e mantido pela empresa Elastic. Trata-se de um servidor de pesquisa baseado na tecnologia Lucene (outro projecto *open-source* da Apache Foundation) orientado a indexar e armazenar dados e facultar capacidades de pesquisa e análise de grandes volumes de dados, de uma forma estruturada e transparente, sobre um

sistema resiliente e com alta disponibilidade (ou tanto quanto as condições provisionadas permitirem).

Em termos de configurações de base, o projecto requer pouca intervenção, já que para colocar uma instância do sistema a correr, basta descarregar um executável do *site* do fabricante e executá-lo com as configurações por omissão. Apesar dessa simplicidade base, são fornecidas capacidades de configuração de elevada granularidade, desde estratégias de replicação dos dados armazenados, a estruturas de processamento de *queries* e até propriedades de descoberta automática dos nós de um *cluster*.

Em termos de escalabilidade, este projecto apresenta grande potencial, sendo o processo de introdução de nós no sistema tão simples como alterar um par de linhas de um ficheiro de configuração em cada nó e executar a aplicação no nó a adicionar. Também no que toca à extensibilidade vemos uma enorme abertura: para além de uma estrutura orientada ao desenvolvimento de *plugins*, a API REST (que é o meio de interacção com a aplicação, padrão), permite a realização de *queries* (ou pedidos) estruturadas e complexas, ou de simples pedidos sem critério de pesquisa.

Os dados são guardados sob a forma de documentos JSON, podendo estes ter estruturas/tipos definidos *a priori* ou em tempo-real (já que esta é a orientação principal do projecto), à medida que os dados são transportados para dentro do sistema e cada campo é indexado. A hierarquia é bastante simples e subdivide-se da seguinte maneira: **índices** que contêm pelo menos um **tipo** de dados, tome-se, como exemplo, o nosso caso em que o "Twitter" é um **índice** e o "*tweet*" um **tipo**, sendo que não é necessária qualquer configuração inicial para que se possam transmitir dados de/para um certo "tipo". Todo o sistema está desenhado para lidar com grandes volumes de dados (*Big Data*) estando, por isso optimizado para o processamento e transmissão de informação por agregado.

Grande parte do processamento é realizado do lado servidor, o que facilita a tarefa das aplicações cliente, que perdem a necessidade de preocupação com a gestão de informação duplicada ou com a manutenção de vistas coerentes sobre os dados contidos na plataforma.

Servem os parágrafos anteriores para introduzir alguma funcionalidade e capacidades do ElasticSearch de uma forma breve e não exaustiva. Neste trabalho optámos por utilizar esse projecto, pela sua simplicidade e versatilidade. Deparámo-nos, ao longo do trabalho com alguns problemas, aquando da configuração do sistema no ambiente de testes em que foi realizada a prova de conceito. Apesar da sua simplicidade base, o ElasticSearch possui inúmeras funcionalidades e configurações de elevada granularidade que o tornam versátil o suficiente para se adequar aos mais diversos ambientes.

A escolha, apesar das dificuldades que enfrentámos, manteve-se, pois o projecto facultou-nos as capacidades necessárias para ocupar uma posição central na solução desenvolvida, tomando o papel do "**motor**", de que já falámos. Para além da facilidade na indexação/introdução de dados no sistema, a simplicidade no uso e instalação de *plugins* para monitorização do *cluster* e visualização dos dados e a capacidade de interagir com o mesmo através de uma API em Python (fornecida também pelo fabricante) e de fazer *queries* simples ou de maior complexidade (como é o caso das *custom_score queries*, de que falaremos mais tarde) tornaram-no uma escolha óbvia.

Na prova de conceito (ou **avaliação** da solução) que veremos na secção 4, focamo-nos nas funcionalidades essenciais de que nos servimos para aferir a aplicabilidade da solução. A realização de pesquisas sobre texto integral ("*full-text search*"), a realização de pesquisas com ponderação de peso definidas por nós ("*custo_score queries*") e a indexação/inserção e extracção de dados através da API base, em aglomerado (*bulk*), são exemplos de capacidades que nos ajudaram a alcançar os objectivos do protótipo.

3.4 *Entrada* a *Saída* de dados no ElasticSearch

Para além dos componentes já mencionados, são também fulcrais para a solução os *scripts* **B**, **C** e **D**. Estes lidam com actividades de importação e exportação de dados de/para o ElasticSearch e de/para a plataforma HP ArcSight.

Os *scripts* **B** e **C** (presentes nos anexos B e C), intitulados, respectivamente, "**outstream_twitter.py**" e "**trainspotter.py**", transportam dados, realizando pesquisas sobre o conteúdo do ElasticSearch e enviando pacotes UDP[5] em formato CEF para um SmartConnector. Este, por sua vez, escuta um porto específico, à espera de pacotes *Syslog* formatados de acordo com o standard CEF para envio para a plataforma ArcSight.

À excepção do volume de dados sobre o qual cada um incide e do tipo de *query* que realizam, não existe diferença entre o funcionamento de cada um. Apesar disso, foi necessário utilizar uma API de Python diferente", designada "**rawes**" [55], no caso em que a *query* utilizada foi uma do tipo "*custom_score"*.

[5] Apesar da falta de garantias no protocolo UDP, a perda pontual de pacotes é, para a maioria dos fabricantes de sistemas SIEM, algo tolerável, tendo em conta a dimensão dos fluxos de dados com que os mesmos lidam. Trata-se de um compromisso entre rapidez (um dos focos destes sistemas é a disponibilidade da informação em tempo-real) e perda de informação.

A execução destes aplicativos é composta dos seguintes passos:

- **Em primeiro lugar**, começam por estabelecer ligação ao servidor do ElasticSearch através da API em Python (no caso do *script* **C** é também extraída a lista de palavras-chave e respectivos pesos, conforme votadas pelos analistas, algo que descrevemos mais adiante);
- **Seguidamente** iniciam um **ciclo** em que vão retirar os dados do índice que lhes é indicado (neste caso, o índice contendo os *tweets*):
    - No caso de **B** a extracção é feita sem critério e, após terem sido retirados todos os documentos (JSON), o aplicativo aguarda 10 segundos até recomeçar a extracção;
    - No caso de **C** é feita uma *custom_score query* que é acompanhada da lista previamente extraída (contendo as palavras-chave e respectivos "pesos"):
        - Graças a esta busca especial, os pesos passados irão sobrepor-se ao mecanismo de pontuação automático do sistema e o peso de cada palavra-chave presente no *tweet* é somado ao peso **total** do *tweet*.
        - Os resultados da pesquisa são devolvidos por ordem decrescente do seu peso **total**.
- **Após essas pesquisas** terem decorrido (ainda dentro do **ciclo** em questão), os dados extraídos são encaminhados para o SmartConnector, sendo, então, passados à plataforma ArcSight no formato CEF.
    - Cada evento (ou documento) contém informações como o utilizador responsável pela publicação, o conteúdo da publicação e o *timestamp* em que esta decorreu (atribuído pelo próprio Twitter).

Existe ainda um quarto aplicativo (**D**) que pode ser considerado um produtor de dados, no entanto, por uma questão de manter a coerência da lógica da aplicação, descrevemo-lo na secção 3.5.

3.5 Integração com o ArcSight

A introdução de dados na plataforma ArcSight é realizada através de um SmartConnector do tipo *Syslog* CEF. No caso concreto do nosso protótipo, a implementação desse componente passou pela instalação de um *daemon*, numa máquina Windows Server 2008, que escuta um porto específico (514 no caso) à espera de pacotes UDP em formato CEF. Recebendo-os ele extrai a informação necessária e encaminha os dados gerados para a plataforma ArcSight *Express* sob a forma de eventos.

Estando os eventos na plataforma, podemos, então, operar sobre eles e criar vários recursos de conteúdo para visualizar a informação recolhida. No ambiente em questão, e como mostrado na figura 13 (à direita), o resultado da extracção de informação do Twitter foi transportado até ao utilizador sob a forma de *Active Channels*, *Dashboards* e *Data Monitors*, e foi manipulado através de **comandos de integração** do tipo "*script*".

3.6 Mecanismo de *feedback*

Construímos um *script* para integrar o *feedback* do analista no ElasticSearch, por forma a apoiar a análise e extracções realizadas, o mesmo encontra-se indicado na imagem como **D** e é intitulado "**voteup.py**" (presente no anexo D). Este componente foi desenvolvido em ambiente Linux e transportado para ambiente Windows, na forma de um executável isolado e sem dependências, com o apoio da ferramenta "*pyinstaller*" [56].

A sua operação realiza um conjunto reduzido de passos para que o analista possa "votar" em conteúdos:

1. O analista selecciona um evento e, com o botão direito do rato, activa o *script* a partir do menu apresentado;
2. O *script* analisa o conteúdo da publicação seleccionada e extrai todas as palavras relevantes, actualizando os pesos das palavras-chave já armazenadas no ElasticSearch;
3. Finalmente o aplicativo é também encarregue de eliminar marcas de pontuação e outros caracteres inúteis (qualquer palavra com menos que 3 letras é descartada) dessa lista.

O sistema possui, à partida, de uma lista de palavras-chave que contém as palavras inicialmente seleccionadas para filtrar a extracção de dados do Twitter e respectivos "pesos". Com o decorrer de votações sucessivas por parte do analista, os conteúdos vão sendo modificados e podemos subdividi-los em dois tipos:

- **Palavras-chave principais** – colocadas na plataforma antes da extracção
    - São catalogadas com o identificador "*main*" e é-lhes atribuído um peso inicial de 5;
    - Cada incremento/voto do analista é contabilizado como um acréscimo de 5 pontos no seu valor.
- **Palavras-chave secundárias** – inseridas ao longo das votações
    - À medida que a ferramenta vai sendo executada e que novas palavras-chave vão sendo "descobertas", elas são inseridas na lista um peso inicial de 2;
    - Cada vez que uma nova votação do utilizador contiver uma dessas palavras e ela já estiver na lista, o valor é aumentado também em 2.

Acreditamos que esta abordagem de aprendizagem de máquinas supervisionada [57], apesar de simplista, nos vai permitir atribuir relevância à informação que suscita maior interesse no analista, já que toda a votação é sustentada na sua observação daquilo que se considera como sendo dados úteis. A introdução de novas palavras (inexistentes, no início da ferramenta) contidas em publicações assinaladas como relevantes levar-nos-á, a que se revele, por inerência, outro conteúdo associado que possa não ter sido observado pelo analista.

Neste capítulo definimos concretamente o funcionamento da nossa ferramenta. Começando por uma síntese da arquitectura e dos vários componentes do protótipo, passámos a detalhar a execução de cada um dos *scripts* e os fluxos de transferência de dados e indexação de *tweets*. Cobrimos também as várias externalidades (como as restrições impostas pela API do Twitter e a necessidade de evitar duplicados e prevenir contra quebras abruptas na execução das aplicações) que definem as nossas opções de implementação. Finalmente, especificamos o comportamento do mecanismo de *feedback* que pretende criar um conjunto de dados que possibilite que a nossa ferramenta "aprenda".

Na avaliação apresentada no próximo capítulo, procuramos evidenciar empiricamente que esta análise, apesar de simplista, obtém resultados satisfatórios. Procuramos que o protótipo seja capaz de, reduzindo o volume de informação a analisar, mantenha a informação relevante, presente no conjunto que é alvo de refinamento. Este resultado apresenta grande valor já que reduz o esforço do analista, automatizando a sua tarefa, tal como nos propusemos fazer, no início do trabalho.

Capítulo 4

Avaliação

Neste capítulo apresentamos uma validação experimental da aplicabilidade da solução desenvolvida. É nosso objectivo, com esta secção, evidenciar que a ferramenta construída constitui um acréscimo de funcionalidade ao SIEM ArcSight e que os resultados apresentados estão coerentes com o que idealizámos à partida.

A secção 4.1 explicita os critérios e processos utilizados para criar a nossa prova de conceito e como foi feita a recolha dos dados. Um sumário dos resultados é apresentado na secção 4.2. Finalmente na secção 4.3 discutem-se esses resultados.

4.1 Metodologia

A figura 14 esquematiza os fluxos de dados que recolhemos para realizar a análise. Na secção 4.1.3 são apresentados os detalhes sobre as várias recolhas. Para fazermos a recolha seguimos o processo seguinte:

1. Recolhemos os *tweets* directamente do Twitter (com base na lista inicial de palavras-chave);
2. Analisámo-los para contabilizar o **total de *tweets* úteis** e o **total extraído**;
3. Refinámos esse **conjunto inicial** de **dados não-refinados** através do nosso protótipo;
4. Com o **resultado** do refinamento, procedemos a nova análise sobre o conjunto de **dados refinados** e contabilizámos, novamente, o **total de *tweets* úteis** e o **total extraído** (ou produzido, neste caso);

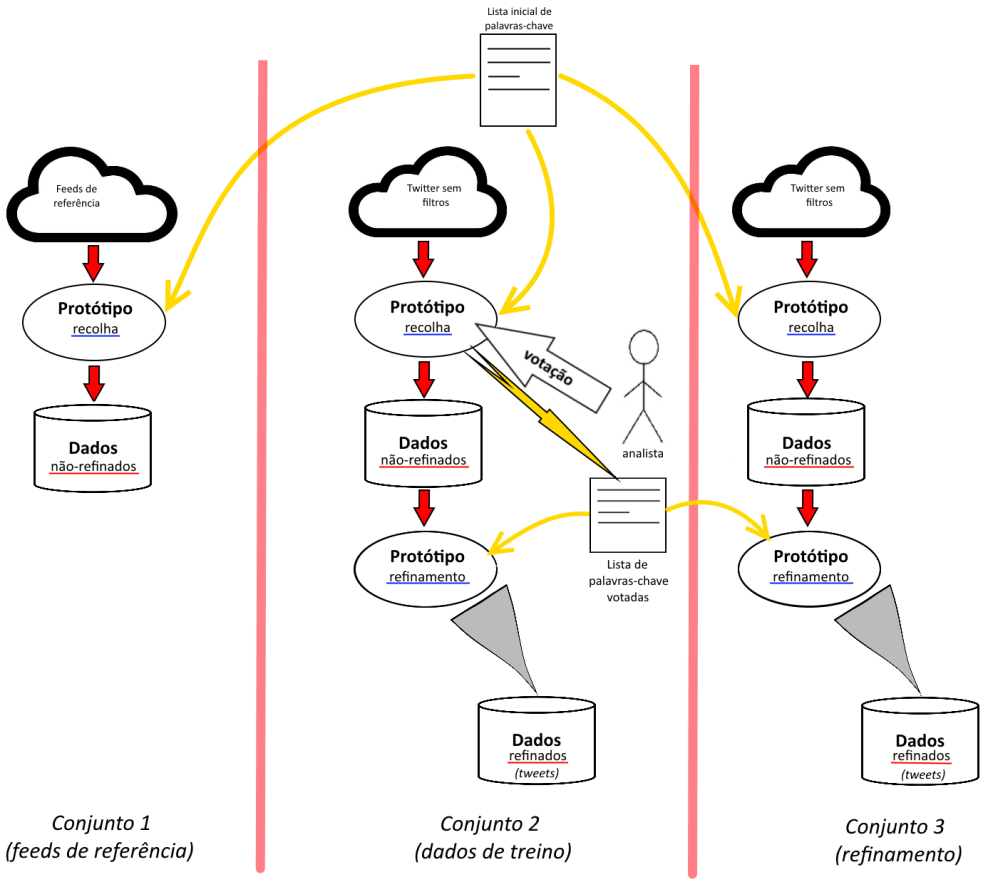

Figura 14: Fluxos de extracção de dados e votação do analista

Na figura podemos observar que existem 3 fluxos, correspondentes a 3 conjuntos de dados diferentes (**conjuntos 1, 2 e 3**). Desses 3 apenas os dois últimos sofrem um processo de refinamento (representado nos passos 2 e 3). Em qualquer dos fluxos partimos de uma recolha sem filtro e dependente apenas da busca que a ferramenta realiza no Twitter, com base nas palavras-chave inicialmente seleccionadas. O resultado desta primeira actividade é um conjunto de dados **não-refinados**, do qual o analista selecciona conteúdos para **votação**. Desta actividade de votação surge uma lista de palavras-chave com novos pesos que alimentar o processo de **refinamento** que a ferramenta executa sobre os dados **não-refinados** anteriores. Finalmente o resultado que desta última actividade surge é um conjunto de *tweets* **refinados**.

4.1.1 Selecção de palavras-chave

Para que se seleccionassem palavras-chave coerentes com o contexto em que as plataformas SIEM são habitualmente utilizadas, começámos por definir, com o apoio da equipa de segurança da Unisys, um cenário em que um analista poderia tirar partido de informação presente em canais OSINT.

Com isto em vista propusemos um "contexto" fictício, simples, em que um determinado centro de operações SOC estaria a monitorizar toda a actividade de segurança de uma organização na qual estivessem presentes as seguintes tecnologias/componentes:

- Postos de trabalho com distribuições corporativas de Windows 7
- Servidores com Sistema Operativo Windows Server 2008 R2
- Servidores com distribuições Linux (Red Hat Enterprise Linux e CentOS)
- *Firewalls* Barracuda Networks
- *Secure Gateways* Blue Coat *ProxySG*
- Instâncias de Base de Dados IBM DB2
- Instâncias de Base de Dados MySQL
- Websites assentes sobre o Wordpress CMS
- Aplicações web a correr sobre o Apache Tomcat
- *Routers* e *switches* Cisco

Feita esta listagem, procedemos, então, à definição de uma lista de palavras-chave para serem recolhidas pela ferramenta desenvolvida. Para isso foi criada (com recurso à experiência dos peritos em segurança da empresa) uma **lista inicial** com 65 palavras-chave (tabela 1).

Após a **lista inicial** estar definida, fizemos com que fosse passada por 6 membros da equipa de segurança da Unisys. Estes, por sua vez, procederam a assinalar as que consideravam, com base na descrição do nosso cenário, como sendo as 30 mais relevantes (isto é, as que consideravam mais passíveis de gerar maior volume de informação relevante aos olhos de um analista). Das mais votadas seleccionámos o top 30, dando origem a uma **lista final** de palavras-chave, apresentada na tabela 2.

Lista inicial		
0-day	denial of service	redirect
administrator	dos	root
apache	exploit	rootkit
barracuda	exposure	scammer
blue coat	forgery	security
botnet	hack	shellcode
breach	hacker	spammer
brute force	hacking	sudo
C&C	injection	threat
cisco	keylogger	threat intelligence
conficker	linux	tomcat
cracker	malware	trojan
cross site	microsoft	virus
cyber attack	misconfiguration	vuln
cyber command	navigation	vulnerability
cyber security	overflow	webapps
cyber terror	phishing	windows
cybersecurity	phreaking	wordpress
db2	port sweep	xss
ddos	portscan	zombie

Tabela 1: Lista de palavras-chave inicialmente seleccionadas

Lista final		
apache	ddos	security
barracuda	denial of service	shellcode
blue coat	dos	threat
botnet	exploit	tomcat
breach	hack	vuln
buffer	injection	vulnerability
cisco	linux	webapps
cross site	malware	windows
cybersecurity	microsoft	wordpress
db2	overflow	Xss

Table 2: Lista de palavras-chave seleccionadas, após deliberação em conjunto com a equipa Unisys

As 30 palavras mais votadas (pós-refinamento) passaram, assim, a servir de base à nossa recolha de dados. Para isso, esta lista foi inserida no ficheiro de configuração do *script* **tweetsniff.py**, em que estão definidas, entre outros atributos, as palavras que vão ser utilizadas na recolha de *tweets*. Esta configuração serve para indicar ao aplicativo quais as palavras-chave pelas quais ele deverá orientar as suas buscas no Twitter. O ficheiro em causa apresenta-se no anexo E.

4.1.2 Relevância da informação

O critério através do qual classificamos um *tweet* como sendo relevante foi alvo de especial atenção e definido também com recurso à experiência da equipa de segurança e sua familiaridade com ambientes SOC. O objectivo aqui foi o de tentar definir concretamente o que distingue um artefacto de informação com valor de um sem valor, para um analista dum centro de operações.

Com base nestes aspectos, definiu-se que seria apropriado considerar como sendo **relevantes** os *tweets* que apresentem qualquer uma das seguintes características:

- Contenham informação de segurança informática (ou que possa ser de importância para algum aspecto de segurança da informação ou de redes) que esteja directamente associada a alguma tecnologia informática concreta - seja dispositivos, *software* ou *hardware*
- Mencione ou possa estar relacionada com uma ameaça que possa comprometer a segurança de uma infra-estrutura de tecnologias da informação (quer se trate de descobertas de *malware* ou vulnerabilidades *0-day*, o lançamento de actualizações de segurança para componentes informáticos e até menções/referências a ameaças humanas como o *hacktivismo* ou ameaças de particulares relacionadas com o espectro informático).

A definição poderá parecer algo lata à partida, no entanto, a observação empírica levou-nos a determinar que, para aumentar o volume de dados útil e possibilitar um melhor refinamento/intervenção do analista em passos sucessores (na votação de conteúdos úteis), seria do interesse do trabalho relaxar o critério e não o limitar ao contexto previamente definido.

4.1.3 Fontes de Dados

Conforme ilustrado na figura 14, a nossa avaliação considera três conjuntos de dados.

Conjunto 1 (*feeds* de referência)

Como forma de estabelecer um ponto de comparação, foram seleccionados 32 *feeds*, isto é, 32 contas de utilizadores reputados no espectro da segurança informática, entre contas que publicam notificações de segurança (descobertas de vulnerabilidades, *pacthes*

de segurança, etc.), contas de personalidades reputadas do sector e contas de organizações, empresas e equipas de segurança de renome.

Estas contas foram seleccionadas com base em informações disponibilizadas por colegas da equipa Unisys que utilizam a rede social Twitter como forma de receber notícias e notificações sobre segurança informática. A tabela 3 lista essas fontes.

***Feeds* de referência**	
https://twitter.com/mathewjschwartz	https://twitter.com/crdflabs
https://twitter.com/nealweinberg	https://twitter.com/USCERT_gov
https://twitter.com/lennyzeltser	https://twitter.com/malwarelu
https://twitter.com/dangoodin001	https://twitter.com/wpvulns
https://twitter.com/dstrom	https://twitter.com/sansappsec
https://twitter.com/securitywatch	https://twitter.com/SANSInstitute
https://twitter.com/Carlos_Perez	https://twitter.com/threatintel
https://twitter.com/taosecurity	https://twitter.com/B_CCENTRE
https://twitter.com/troyhunt	https://twitter.com/virustotal
https://twitter.com/schneierblog	https://twitter.com/alienvault
https://twitter.com/mikko	https://twitter.com/teamcymru
https://twitter.com/briankrebs	https://twitter.com/TenableSecurity
https://twitter.com/vuln_lab	https://twitter.com/HiveData
https://twitter.com/exploitdb	https://twitter.com/ThreatStream
https://twitter.com/DavinsiLabs	https://twitter.com/TrustedSec
https://twitter.com/OSVDB	https://twitter.com/owasp

Tabela 3: Feeds utilizados como referência na análise

Sobre este conjunto de *feeds* foi analisado o período de tempo de 4 a 11 de Maio de 2015 e foram calculados os resultados sobre a qualidade da informação recolhida, para comparação.

Este conjunto de dados serve apenas para podermos observar a qualidade do refinamento da nossa ferramenta. Os conjuntos refinados que a mesma irá produzir serão comparados com os dados originários das contas indicadas na tabela 3. Com esta análise poderemos estabelecer um paralelismo entre a capacidade do nosso protótipo de refinar informação ao ponto de a tornar tão útil quanto a que é proveniente de um universo, reduzido, de utilizadores focados exclusivamente na área da segurança informática. A figura 14 ilustra os vários processos de extracção de *tweets*.

Conjunto 2 (dados de treino)

Como universo de análise e com base nos critérios e métricas acima mencionados, procedemos à recolha de dados, **de todo o Twitter**, durante o período de **uma semana**,

entre os dias 4 e 11 de Maio de 2015. A ferramenta foi colocada "à escuta", recolhendo (através dos mecanismos de busca) todos os dados referentes às 30 palavras-chave seleccionadas (tabela 2).

Conjunto 3 (de refinamento)

Para validar o mecanismo de *feedback* servimo-nos do **conjunto 3**, recolhido entre os dias 2 e 3 de Julho. Este terceiro conjunto vai ser refinado com base em *feedback* que o utilizador forneceu ao visualizar o **conjunto 2** e surge para que possamos entender como a plataforma se vai comportar com a passagem do tempo.

Numa utilização normal, a plataforma seria colocada num centro SOC e estaria a recolher e refinar dados continuamente, estando os seus utilizadores a "votar" conteúdos periodicamente. Aqui decidimos medir o impacto que a nossa solução sofre quando passado alguns dias o contexto das ameaças e da envolvência, nas redes sociais, se alterou.

4.1.4 *Feedback* do analista

Ao longo do período de tempo de extracção do **conjunto 2**, o autor deste trabalho agiu como analista de segurança e foi "votando" em *tweets* considerados relevantes (segundo os critérios definidos na secção 5.1.2), servindo-se do utilitário de votação descrito na secção 4.6. Esta actividade encontra-se representada na figura 14 (ver o início da secção 4.1).

A análise e escolha foram feitas nas instalações da Unisys, num ambiente análogo ao de um SOC e, durante esse período, o analista foi responsável por utilizar os recursos de conteúdo criados no ArcSight para tentar encontrar informação relevante e assinalá-la para votação. No total foram seleccionados **208** *tweets* como relevantes.

O objectivo principal de toda a análise é o de comparar as percentagens de informação relevante antes e depois do refinamento levado a cabo pela ferramenta. Passamos, depois, a fazer uma comparação entre a densidade de informação relevante no conjunto refinado e a densidade de informação relevante contida nos *feeds* referência (entenda-se, o nosso conjunto de comparação).

4.1.5 Métricas

Quanto ao processo de refinamento automático, decidimos considerar duas métricas para aferir a sua capacidade de adicionar valor e cumprir os nossos requisitos [58]: *accuracy* (ou precisão) e *completeness* (ou completude).

Estas métricas são basilares na avaliação da qualidade de sistemas IDS (*Intrusion Detection* System) e, pela sua relevância dentro do espectro da segurança informática, decidimos adoptar e transportá-las para o nosso paradigma. Para isso, definimo-las da seguinte forma: a primeira, **precisão**, como sendo referente à capacidade do processo de auferir um resultado positivo, reduzindo o volume de dados a tratar e aumentando, em simultâneo, a percentagem/proporção de dados úteis do conjunto analisado; e a segunda, **completude**, como estando associada à maneira como o processo consegue gerar um subconjunto refinado que mantenha o máximo possível dos dados úteis do conjunto inicialmente extraído.

No caso da nossa avaliação, o processo de refinamento (ou pontuação) foi configurado para devolver apenas os **5%** de *tweets* com **maior relevância**, do total de dados extraídos directamente do Twitter (os dados retirados na primeira iteração, nos fluxos da figura 14).

4.2 Resultados

Os resultados da recolha dos vários conjuntos são consolidados na tabela 4. Agrupamos os resultados em duas fases da análise: **pré-refinamento** e **pós-refinamento**, isto é, as duas actividades que recolhem dados do Twitter sem qualquer filtro (para além das palavras-chave seleccionadas) e as que refinam os dados previamente recolhidos, através do mecanismo de *feedback* do utilizador. Em ambas as fases contabilizamos os valores totais de conteúdo extraído, de conteúdo útil e apresentamos um valor para a densidade de informação útil.

	Pré-refinamento			**Pós-refinamento**		
	Total sem filtro (*tweets*)	**Total útil (*tweets*)**	**Informação útil**	**Total sem filtro (*tweets*)**	**Total útil (*tweets*)**	**Informação útil**
Conjunto 1	1097	537	48,95%	-	-	-
Conjunto 2	739656	19713	2,67%	36982	19376	52,39%
Conjunto 3	239074	5799	2,43%	11953	5270	44,09%

Tabela 4: Comparativo de dados e dados úteis extraídos dos vários conjuntos

Como podemos ver, o **conjunto 1** comporta uma boa densidade de informação útil, apesar do volume de conteúdo não ser significativo. Ao alargarmos o âmbito da análise para além dos *feeds* de referência, no entanto, temos que ter em conta que o aumento do volume de informação se traduz, também, no aumento de informação irrelevante. Assim sendo, o refinamento de dados é necessário caso queiramos tomar esse passo.

Na tabela 4 apresentam-se, também, os resultados das extracções dos **conjuntos 2 e 3**. O **conjunto 2** foi recolhido entre os dias 4 e 11 de Maio (período durante o qual o

analista votou nos *tweets* que considerou relevantes) e foi posteriormente refinado pelo nosso protótipo. O **conjunto 3** foi recolhido entre os dias 2 e 3 de Julho e foi refinado com base nos dados de *feedback* do analista, provenientes da análise do **conjunto 2**.

Convém salientar que qualquer dos conjuntos de dados refinados comporta apenas **5%** do conjunto inicial, **não-refinado**. Esta extracção é simples, pois a ferramenta devolve os resultados pontuados e por ordem decrescente de peso. Assim, a análise contempla apenas os primeiros **5%** de resultados (com maior peso).

A figura 15 apresenta um gráfico com as comparações dos vários conjuntos, quanto à métrica de **precisão**. À excepção do **conjunto 1**, cujos dados não são refinados, os **conjuntos 2 e 3** apresentados referem-se aos resultados contabilizados após a actividade de refinamento.

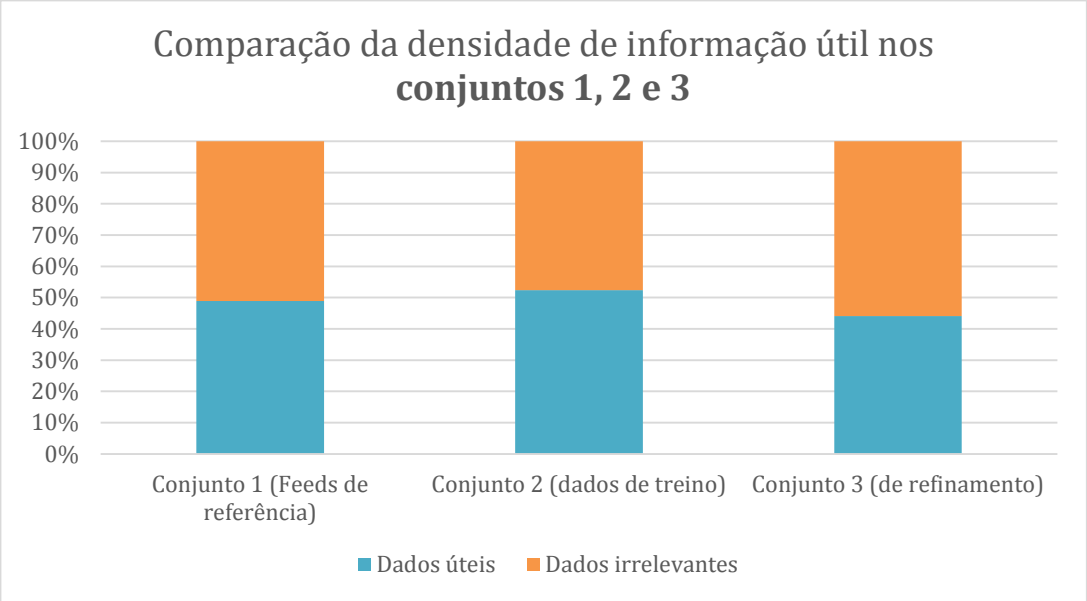

Figura 15: Comparação das percentagens de informação útil recolhida nos conjuntos refinados 1, 2 e 3

No que toca à **precisão** do refinamento sobre o **conjunto 2,** os valores apresentados traduzem-se num aumento de **2,67%** para **52,39%** (cerca de **49,72%**), da percentagem de informação útil dos dados recolhidos, acompanhado da redução do volume de informação que se verificou.

Quanto ao **conjunto 3** podemos observar um aumento da percentagem de informação útil de **2,43%** para **44,09%** (cerca de **41,66%**).

A figura 16 apresenta uma comparação da **completude** da actividade de refinamento dos **conjuntos 2 e 3**, já que no caso do **conjunto 1** essa questão não se coloca.

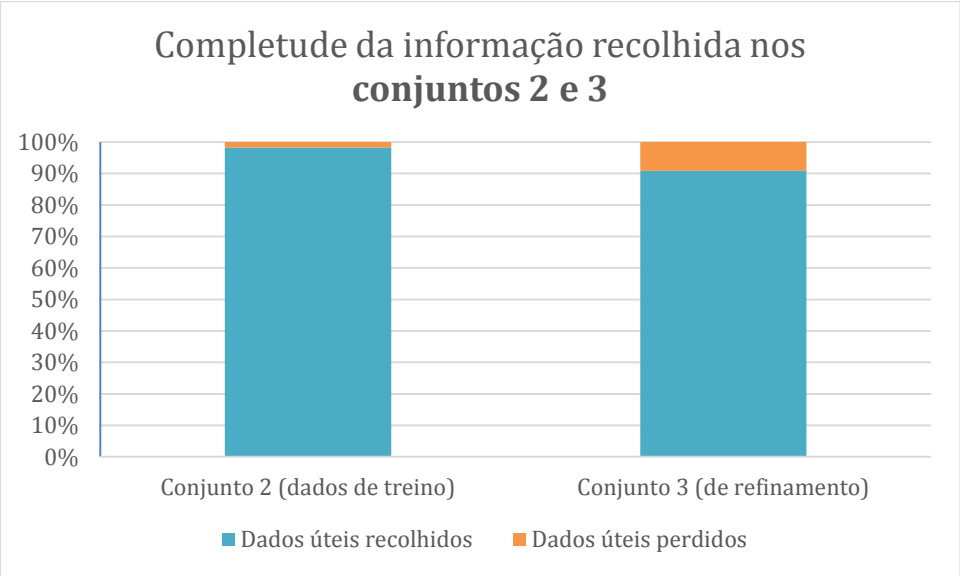

Figura 16: Comparação da completude da informação recolhida pós-refinamento nos conjuntos 2 e 3

Como podemos observar os valores indicam-nos que, no refinamento dos dados do **conjunto 2**, a solução foi capaz de recolher cerca de **98,29%** de toda a informação útil existente no conjunto total de dados extraídos originalmente (não-refinados).

No caso do **conjunto 3** foram recolhidos aproximadamente **90,88%** do conteúdo útil existente no total dos dados extraídos à partida (não-refinados).

4.3 Discussão dos resultados

Os resultados apresentados na secção anterior mostram que a ferramenta que construímos apresenta uma boa capacidade de refinar os dados provenientes da rede social Twitter. Além disso indicam que a actividade de refinamento aumenta a percentagem de informação útil no conjunto de dados refinados em **49,72%** e **41,66%**, em relação aos conjuntos 2 e 3, respectivamente. Já no que toca ao aspecto da completude, podemos observar que no refinamento dos conjuntos 2 e 3 se mantêm **98,29%** e **90,88%** do conteúdo útil, respectivamente. Esta progressão nos resultados indica-nos que, apesar do contexto se ter alterado e as análises incidirem sobre períodos diferentes, a ferramenta continuou a ser capaz de produzir um bom resultado.

Em termos da informação que é perdida e da deterioração do refinamento, identificamos que houve decréscimos na **precisão** (de aproximadamente **8,06%**) e na **completude** (de cerca de **7,41%**). Acreditamos que isto se deve principalmente ao facto de, na semana extraída pelo **conjunto 2**, ter sido descoberta uma vulnerabilidade crítica no CMS Wordpress. Esse evento levou a que a votação do analista fosse influenciada e que palavras associadas a esse tema fossem relevadas em excesso. Uma boa parte dos

resultados do refinamento do **conjunto 3** continham informação irrelevante referente ao CMS Wordpress.

Com esta informação denotamos a necessidade e o impacto positivo que teria tido um mecanismo de votar negativamente um resultado, ou de realizar uma purga sobre os conteúdos votados pelo analista. Se, de alguma maneira, o processo de aprendizagem supervisionada fosse capaz de detectar que o Wordpress tinha deixado de ser tão relevante no período de recolha do conjunto 3, acreditamos que a qualidade da análise se teria mantido ainda mais próxima.

Com efeito, convém também salientar que, o refinamento do conjunto 3 não foi acompanhado da actividade de votação do analista. Mais uma vez, acreditamos que este se trata de um factor condicionante da qualidade dos resultados. Se tal análise tivesse sido feita, acreditamos que o contexto se teria adaptado gradualmente.

Tendo em conta os objectivos iniciais, podemos argumentar que existe um acréscimo de valor na utilização da nossa solução. Apesar da perda de alguma informação útil no processo de refinamento, a redução significativa do conjunto inicialmente extraído do Twitter é muito importante no nosso contexto. O facto de conseguirmos reduzir grandes volumes de dados a conjuntos mais manuseáveis de informação para transportar para o SIEM é fulcral para que a solução tenha aplicabilidade.

Conseguimos, assim, atingir o principal objectivo de integrar informação OSINT (o caso desta avaliação, reduzida ao Twitter) nas plataformas SIEM (ArcSight, para o efeito), de forma automática e com refinamento dos dados extraídos, para facilitar a sua utilização/análise.

Notamos, no entanto, que o foco inicial do trabalho não era o de aprofundar conhecimentos sobre técnicas de aprendizagem de máquinas. Esta necessidade surgiu quando, face ao volume de informação inútil presente nas redes sociais, nos vimos forçados a procurar alguma técnica que reduzisse o volume de informação e aumentasse a sua qualidade. Acreditamos que existe ainda muito espaço para evolução desses mecanismos através do uso de algoritmos mais elaborados.

Capítulo 5

Conclusão e Trabalho Futuro

5.1 Conclusão

Neste trabalho propusemos constituir uma *framework* para suportar uma ideia, na qual reconhecemos bastante valor, face à realidade que o mercado nos apresenta. De facto uma ferramenta capaz de facilitar a tarefa de um analista de segurança apresenta muito potencial, já que reduz a carga que é colocada sobre o analista, libertando-o para que possa desempenhar outras tarefas de criticidade elevada.

O sistema que desenvolvemos tira partido das redes sociais para extrair informação que pode, eventualmente, levar um analista a detectar ameaças e a prevenir e mitigar as capacidades das mesmas. Para esse efeito criámos uma ferramenta que extrai dados do Twitter, refina-os com base numa técnica de aprendizagem de máquinas supervisionada (a abordagem da nossa prova de conceito é bastante simplista), em que o mecanismo de *feedback* provém de informação fornecida pelo utilizador e envia-os para o SIEM ArcSight, para que possam ser analisados.

Face ao objectivo inicial, não só constituímos uma prova de conceito que demonstra a aplicabilidade da solução proposta e desenvolvida, como criámos um trabalho que consideramos percursor e passível de melhoria no futuro, estabelecendo uma base daquilo que vai sendo cada vez mais fundamental na área da analítica de segurança. Apesar de simplistas, as nossas técnicas representam um acréscimo de valor às actividades que a ferramenta poderá suportar, e são simbólicas do potencial que pode ter um trabalho subsequente com foco na parte da nossa ferramenta associada a reduzir e refinar os conjuntos de dados em análise.

Para além do esforço de investigação que nos levou até este ponto (e a reconhecer uma forma de colmatar a necessidade identificada), desenvolvemos e testámos, com resultados positivos, um protótipo com uma utilização que pode ter grande utilidade para os clientes da própria empresa em que o estágio se insere.

5.2 Discussão

Tendo todo o trabalho de revisão do comportamento da ferramenta sido manual, pudemos observar alguns aspectos específicos da análise que não transparecem através

da estatística. Um desses factos prende-se com mais de 70% dos resultados úteis presentes no conjunto pós-refinamento estarem contidos nos cerca de 50% de resultados com maior pontuação do conjunto (ou seja, são os primeiros "da lista").

Não desconsideramos a relevância que poderá ter a informação que é perdida pela ferramenta após a actividade de refinamento. Em centenas de *tweets* podem estar as evidências mais claras de ameaças a uma organização, no entanto, trata-se de um compromisso que deve ser entendido como um risco e cabe a cada um adequá-lo ao seu nível de confiança. Note-se que o valor de 5% de informação extraída dos conjuntos não-refinados foi definido com base em observação empírica de que nesse subconjunto estaria a maior parte da informação útil.

O mecanismo de pontuação baseado em *feedback* do utilizador (analista, neste caso) apresenta um resultado positivo. No entanto, podemos especular sobre uma utilização a longo prazo em que o conteúdo da lista da votação (lista de palavras-chave que vão ganhando relevância consoante a votação do utilizador) vai crescendo e sendo aprumado. Aqui podem surgir vários cenários díspares: não só esta lista pode crescer exponencialmente e a carga de processamento ser tão elevada que a ferramenta acaba por ser impossibilitada de produzir resultados em tempo útil; mas também, na ausência de um mecanismo de purga de informação irrelevante, o conteúdo da mesma poderá tornar-se repetitivo ou comportar elementos que não acrescentam valor, ou até, deterioram a capacidade analítica da ferramenta (suponha-se o caso de ganharem relevância excessiva palavras que, por serem comuns na própria língua, aparecem muitas vezes em publicações de utilizadores). Neste aspecto, apontamos o mecanismo de refinamento como merecedor de especial atenção, quer através da adição de um mecanismo de purga, ou que retire valor/pontuação/importância a palavras-chave irrelevantes no âmbito em caso, ou da introdução de métodos mais complexos de refinamento de texto "útil".

Ainda quanto ao volume de dados produzido devemos salientar que, apesar da significativa redução do conjunto de informação que a ferramenta foi capaz de realizar, não podemos descartar o facto de vários milhares de frases/parágrafos serem de difícil análise, mesmo para um analista experiente. Aqui achamos ser essencial, ainda no âmbito da plataforma, a utilização de técnicas de visualização que sejam capazes de apresentar dados textuais (associados a comunicação de pessoas) de forma útil e comfortável para um analista, algo não suportado pelas plataformas SIEM actualmente em uso.

5.3 Experiência

O trabalho que aqui se apresenta foi desenvolvido em simultâneo com outras actividades levadas a cabo pelo aluno na Unisys, de manutenção e implementação da

plataforma ArcSight, desenvolvimento de conteúdos para a mesma e realização de auditorias de segurança. Especificamente ao desenvolvimento do trabalho foi alocado cerca de 40% do tempo do aluno na empresa.

O processo de estágio curricular apresentou uma série de benefícios para o trabalho aqui apresentado, tendo-se, graças a ele, e à colaboração da Unisys, ganho uma melhor compreensão do estado-de-coisas na área do SIEM e da segurança informática em geral.

A participação activa dos restantes colegas da empresa, a convivência com os clientes e o acesso a equipamento actual para condução da avaliação da ferramenta facilitaram bastante o trabalho. Para além do trabalho desenvolvido, as actividades desempenhadas pelo aluno passaram, essencialmente, pela implementação, gestão e manutenção da plataforma ArcSight em 3 clientes de diferentes sectores (projectos que foram construídos de raiz e que contemplaram vários esforços de integração). Adicionalmente o aluno esteve envolvido num projecto internacional com uma grande empresa do sector dos serviços financeiros, desenvolvendo conteúdo para utilização por parte de um SOC.

5.4 Trabalho Futuro

Acreditamos existirem 4 oportunidades de melhoria do trabalho aqui apresentado.

Em primeiro lugar, e sendo um dos pontos onde poderá existir grande margem de melhoria, colocamos a **extracção de dados** de mais fontes. Na prova de conceito apresentada, optámos por nos focar, essencialmente por limitações temporais e pela diversidade de componentes que a solução exigiu, na recolha de informação da rede social Twitter. No entanto, a recolha pode (e acreditamos que deve) ser alargada a qualquer canal de informação OSINT que possa ser transformado em fluxos de informação textual, quer se trate de fóruns, abertos ou "fechados" (muitos fóruns exigem registo de conta para visualização de conteúdo), outras redes sociais e até meios mais esotéricos (do ponto de vista do processamento de informação textual) como chamadas telefónicas (por exemplo, dentro do perímetro da organização, para detectar fugas de informação) ou SMSs. Esta adição de fontes vai não só substanciar os conjuntos de informação disponível para análise, dando maior visibilidade sobre o cômputo geral da informação em circulação, como também possibilitar o cruzamento da informação disponível de fontes distintas e enriquecer a análise da(s) ameaça(s) em causa.

Em segundo lugar, e ocupando uma posição central na nossa *framework*, passível de significativa melhoria, observamos o nosso instrumento de **refinamento de dados**. Aqui parece-nos evidente que existe margem para que o mecanismo de votação seja refinado

de forma a colmatar os problemas revelados na conclusão do trabalho. O objectivo principal deste componente era o de criar algo que fosse capaz de, seguindo orientações de um analista de segurança de um SOC, pré-calcular a informação que este poderá entender como sendo relevante e descartar o restante conteúdo. Problemas como a degradação do desempenho da análise com o aumento da intervenção do analista e a possibilidade de se danificar a análise devido ao método simplista utilizado (no caso de não existirem medidas de prevenção desse facto) levam-nos a considerar outros processos de refinamento de conjuntos de dados.

Sobre esse mesmo tema surgem abordagens muito interessantes e, neste momento, estamos a considerar um complemento ao método da solução, recorrendo a mecanismos de *análise de agrupamentos* [59]. O uso desses mecanismos pode facilitar ao utilizador a detecção de situações anómalas ou inesperadas, através do agrupamento, por exemplo, de publicações com algum indicador de relevância (que representem risco potencial de ameaça), em conjuntos similares. Isto é algo que pode ser muito interessante na detecção e resposta a ameaças de *hacktivismo* que tentem dispersar informação e reunir apoiantes através das redes sociais, ou mesmo na prevenção de dispersões de *malware* (através de alertas propagados/publicados nas redes sociais e que se tornam quase "virais", como foi o caso do *Rombertik* [61], aquando da avaliação da ferramenta).

Ainda quanto a abordagens de aprendizagem automática, estamos a planear uma segunda rota de melhoria da solução, contemplando uma tentativa de *association analysis* (análise de associações), mais concretamente, *association rule mining* [60] (extracção de regras de associação)**,** para possibilitar a descoberta de padrões de correlação entre os dados analisados que possam ser indicativos de actividade maliciosa.

Um ponto também muito relevante é o da **visualização** da informação textual recolhida. Métodos eficazes de transmitir conhecimento ao analista, através de técnicas de visualização de dados, especificamente textuais e de comunicação interpessoal, são escassos no âmbito das plataformas SIEM, por isso acreditamos estar aqui um ponto merecedor de atenção num trabalho futuro.

Finalmente o terceiro principal ponto de melhoria da nossa solução prende-se com a integração da solução com plataformas SIEM, ou o **tradutor** (***wrapper***), como denominado na definição da arquitectura. Aqui as melhorias são de menor complexidade e bastante mais orientadas a possibilitar o funcionamento com outro tipo de SIEMs. Reconhecemos a importância, ainda que não ao nível da dos restantes pontos, de tornar esta ferramenta o mais aberta e multiplataforma possível, para facilitar a sua utilização em ambientes reais, diversos.

Glossário

API

Application Programming Interface – conjunto de ferramentas para auxiliar os desenvolvedores de *software* na criação de aplicações

arcOSI

Software open-source para extracção de dados de fontes abertas e transporte para o ArcSight

ArcSight

Solução de SIEM empresarial, desenvolvida e mantida pela empresa Hewllet-Packard

Big Data

Termo que representa recolha de dados (e possivelmente análise e extracção de informação) de grandes conjuntos, tão volumosos, que geralmente requerem técnicas especializadas para serem manuseados

Business Intelligence

Conjunto de técnicas para transformar dados do negócio em informação (e conhecimento) relevante à actividade de uma organização, potenciando a tomada de decisões informada e omnisciente (em aspectos do negócio)

CEF

Common Event Format – o formato através do qual o ArcSight processa os eventos/dados dentro do sistema

Commercial Intelligence

Informação de fontes abertas e/ou obtida através de meios legítimos e que pode revelar dados comerciais de grande relevância que contribuam positivamente para auxiliar uma

organização a atingir os seus objectivos. Pode ser dividida em Business Intelligence, e Competitor/Competitive Intelligence

Competitor/Competitive Intelligence

Conjunto de actividades associadas à recolha e utilização de informação em produtos, clientes ou competidores

CRM

Customer Relationship Management – sistema (ou conjunto de sistemas) para gestão das interacções de uma organização com os seus clientes

Extracção de Informação

Processo de descoberta de padrões, detecção de informação estratégica e derivação de conhecimento a partir de dados contidos em conjuntos muito grandes

Firewall

Dispositivo para mediar e controlar o acesso a recursos de informação, interpondo-se entre a máquina/dispositivo que pretende obter acesso a um determinado recurso e o próprio recurso

Hacktivist

Um utilizador mal-intencionado que procura corromper, comprometer, atacar ou destruir activos de informação, pertencentes a uma empresa ou organização, motivado por crenças ou razões políticas

IAM

Identitity and Access Management – paradigma e conjunto de ferramentas/produtos para a gestão de identidades de conjuntos de utilizadores de sistemas, e que pode controlar/mediar processos de autenticação e autorização em sistemas ou aplicações

IDS

Intrusion Detection System – dispositivo que monitoriza redes e/ou sistemas, em busca de evidências de comprometimento ou actividade maliciosa, e que pode ter capacidades de despoletar alertas ou acções correctivas

Aprendizagem de máquinas

Área da inteligência artificial que estuda técnicas que permitam aos computadores desenvolver capacidades de aprendizagem e dedução de informação

Processamento de Linguagem Natural

O estudo e técnicas associadas à compreensão da linguagem humana, com capacidades para inferência/dedução de significado ou intenção (entre outras)

Open-Source

Condição daquilo que é aberto, gratuito e de livre uso por qualquer individuo, em qualquer altura

OSCINT

O mesmo que OSINT

OSINT

Open-Source Intelligence – Informação aberta e livremente disponível, que flui em canais de livre acesso por parte de qualquer individuo

Script-Kiddy

Um individuo pouco instruído e pouco dotado que se serve de scripts ou programas desenvolvidos por terceiros para tentar atacar sistemas ou infra-estruturas

SEM

Security Event Management – conceito e conjunto de ferramentas que definem actividades associadas à recolha e armazenamento de informação de segurança, focado no armazenamento a longo prazo, para efeitos de facilitar a análise histórica

SIEM

Security Information and Event Management – conceito e conjunto de ferramentas que combinam as capacidades dos sistemas SIM e SEM

SIM

Security Information Management – conceito e conjunto de ferramentas que definem as actividades relacionadas com a recolha e processamento de informação de segurança, focado na disponibilização de dados em tempo-real

SOC

Security Operations Centre – centro para protecção, monitorização e controlo de activos de informação

Redes Sociais

Plataformas online para facilitar a comunicação entre indivíduos ou os seus "pseudónimos" virtuais. O termo é considerado no contexto da Internet

Bibliografia

[1] D. R. Miller, S. Harris, A. Harper, S. VanDyke e C. Blask, "Security Information and Event Management (SIEM) implementation," D. R. Miller, Ed., McGraw-Hill, 2011, pp. XXV-XXXIV.

[2] L. Hewlett-Packard Development Company, "Life Cycle of an Event Through ESM," em *ESM 101, Concepts for ArcSight ESM 6.5c SP1*, Hewlett-Packard Development Company, L.P., 2014, p. 27.

[3] E. J. Appel, Cybervetting: Internet Searches for Vetting, Investigations, and Open-Source Intelligence, Crc Press, 2014.

[4] S. L. Technology, *SAIL LABS Technology :: ROSIDS.*

[5] A. Williams, "The Future of SIEM--The market will begin to diverge," *Retrieved,* vol. 12, nº 1, p. 2011, 2007.

[6] M. Nicolett e K. M. Kavanagh, "Magic quadrant for security information and event management," *Gartner RAS Core Reasearch Note (May 2009),* 2014.

[7] X. Mertens, "SIEM-architecture," 2007. [Online]. Available: http://blog.rootshell.be/2007/06/22/you-said-siem/siem-architecture/.

[8] A. Miller, "ArcSight Architecture," 2013. [Online]. Available: http://www.allymiller.info/blog/risk/2013/08/376/.

[9] L. Hewlett-Packard Development Company, "ESM Anatomy," em *ESM 101, Concepts for ESM 6.0c with CORR-Engine*, Hewlett-Packard Development Company, L.P., 2014, pp. 19-30.

[10] L. Hewlett-Packard Development Company, "User's Guide. ArcSight Console," Hewlett-Packard Development Company, L.P., 2014.

[11] R. D. Steele, The new craft of intelligence, OSS International Press, 2002.

[12] R. D. Steele, "The importance of open source intelligence to the military," *International Journal of Intelligence and Counter Intelligence,* vol. 8, nº 4, pp. 457-470, 1995.

[13] R. D. Steele, "Open source intelligence: What is it? why is it important to the military," *American Intelligence Journal,* vol. 17, nº 1, pp. 35-41, 1996.

[14] H. Bean, No More Secrets: Open Source Information and the Reshaping of US Intelligence: Open Source Information and the Reshaping of US Intelligence, ABC-CLIO, 2011.

[15] A. Olcott, Open source intelligence in a networked world, vol. 7, A&C Black, 2012.

[16] C. Fleisher, "OSINT: Its Implications for Business/Competitive Intelligence Analysis and Analysts," *Inteligencia y Seguridad,* vol. 2008, nº 4, pp. 115-141, 2008.

[17] R. Gupta e H. Brooks, Using Social Media for Global Security, John Wiley & Sons, 2013.

[18] A. Al Hasib, "Threats of online social networks," *IJCSNS International Journal of Computer Science and Network Security,* vol. 9, nº 11, pp. 288-93, 2009.

[19] Michael, *Some Open Source Intelligence Basics,* 2012.

[20] J. M. Kleinberg, "Challenges in mining social network data: processes, privacy, and paradoxes," em *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007.

[21] M. Zuber, "A Survey of Data Mining Techniques for Social Network Analysis," *International Journal of Research in Computer Engineering \& Electronics,* vol. 3, nº 6, 2014.

[22] M. Atzmueller, "Data mining on social interaction networks," *arXiv preprint arXiv:1312.6675,* 2013.

[23] M. M. Rahman, "Mining Social Data to Extract Intellectual Knowledge," *arXiv preprint arXiv:1209.5345,* 2012.

[24] A. Amine, R. M. Hamou e M. Simonet, "Detecting Opinions in Tweets," *International Journal of Data Mining and Emerging Technologies,* vol. 3, nº 1, pp. 23-32, 2013.

[25] T. Wang e H. Zhu, "Voting for Deceptive Opinion Spam Detection," *arXiv preprint arXiv:1409.4504,* 2014.

[26] S. Kampakis e A. Adamides, “Using Twitter to predict football outcomes,” *arXiv preprint arXiv:1411.1243,* 2014.

[27] P. Domingos, “Mining social networks for viral marketing,” *IEEE Intelligent Systems,* vol. 20, nº 1, pp. 80-82, 2005.

[28] L. Tabourier, A.-S. Libert e R. Lambiotte, “RankMerging: Learning-to-rank in large-scale social networks (extended),” *CoRR,* vol. abs/1407.2515, 2014.

[29] W. Magdy, H. Sajjad, T. El-Ganainy e F. Sebastiani, “Bridging Social Media via Distant Supervision,” *CoRR,* vol. abs/1503.04424, 2015.

[30] C. Fang, H. Jin, J. Yang e Z. L. Lin, “Collaborative Feature Learning from Social Media,” *CoRR,* vol. abs/1502.01423, 2015.

[31] Nimble, “CRM,” Nimble, Inc, 2015. [Online]. Available: http://www.nimble.com/.

[32] Salesforce, “CRM,” Salesforce.com, 2015. [Online]. Available: http://www.salesforce.com.

[33] Zoho, “CRM,” Zoho, Inc, 2015. [Online]. Available: https://www.zoho.com/.

[34] I. Salesforce.com, “Getting Started with Social Customer Service,” 2015. [Online]. Available: https://help.salesforce.com/help/pdfs/en/social_customer_service_impl_guide.pdf.

[35] R. Jurdak, A. Elfes, B. Kusy, A. Tews, W. Hu, E. Hernández, N. Kottege e P. Sikka, “Autonomous surveillance for biosecurity,” *CoRR,* vol. abs/1503.01173, 2015.

[36] J. N. Matias, A. Johnson, W. E. Boesel, B. Keegan, J. Friedman e C. DeTar, “Reporting, Reviewing, and Responding to Harassment on Twitter,” *CoRR,* vol. abs/1505.03359, 2015.

[37] W. Gharibi e M. Shaabi, “Cyber threats in social networking websites,” *CoRR,* vol. abs/1202.2420, 2012.

[38] A. Guille e C. Favre, “Event detection, tracking, and visualization in Twitter: a mention-anomaly-based,” *Social Netw. Analys. Mining,* vol. 5, nº 1, pp. 18:1--18:18, 2015.

[39] B. A. Miller, M. S. Beard e N. T. Bliss, “Eigenspace analysis for threat detection in social networks,” em *Proceedings of the 14th International Conference on Information Fusion,*, 2011.

[40] S. Dilek, H. Çakir e M. Aydin, “Applications of Artificial Intelligence Techniques to Combating Cyber,” *CoRR*, vol. abs/1502.03552, 2015.

[41] A. R. Zope, A. Vidhate e N. Harale, “Data minding approach in security information and event management,” *J. Future Comput. Commun*, 2013.

[42] P. a. D. G. Maciolek, “{CLUO:} Web-Scale Text Mining System for Open Source Intelligence Purposes,” *Computer Science {(AGH)}*, vol. 14, nº 1, pp. 45-62, 2013.

[43] I. M. De Santos e A. M. Vega, “Las fuentes abiertas de información. Un sistema de competencia perfecta,” *Inteligencia y seguridad: Revista de an{\'a}lisis y prospectiva*, nº 8, pp. 91-112, 2010.

[44] M. Karaman e H. Çatalkaya, “Institutional Cybersecurity: A Case Study of Open Source Intelligence and Social Networks”.

[45] D. Gritzalis, “Open-Source Intelligence produced from Social Media: A proactive cyber defense tool,” 2014.

[46] I. McAfee, “McAfee ZeroFOX Solution Brief,” 2015. [Online]. Available: https://www.zerofox.com/wp-content/uploads/z/McAfeeZF-Integrations-Brief.pdf.

[47] I. ThreatConnect, “ThreatConnect,” 2015. [Online]. Available: http://www.threatconnect.com/news/hiding-in-the-clouds/.

[48] I. Cyveillance, “Cyveillance,” [Online]. Available: https://www.cyveillance.com/.

[49] I. Soteria, “Soteria,” 2015. [Online]. Available: http://www.soteriaintelligence.com/social-media-threats/.

[50] G. Martin, “ArcOSI,” 2011. [Online]. Available: https://code.google.com/p/arcosi/.

[51] Unknown, “Collective Intelligence Framework,” 2014. [Online]. Available: https://code.google.com/p/arcosi/.

[52] A. M. Swanda, "ArcReactor," [Online]. Available: http://deadbits.org/projects/arcreactor/.

[53] I. ThreatStream, "ThreatStream OPTIC," 2015. [Online]. Available: https://www.threatstream.com/platform/threatstream-integrator.

[54] Elastic, "ElasticSearch Docs," 2015. [Online]. Available: https://www.elastic.co/guide/index.html.

[55] X. Mertens, "tweetsniff," GitHub, 2015. [Online]. Available: https://github.com/xme/tweetsniff.

[56] HumanGeo, "rawes - Low level elasticsearch driver for Python," [Online]. Available: https://github.com/humangeo/rawes.

[57] PyInstaller, "PyInstaller - PyInstaller official GIT repository," [Online]. Available: https://github.com/pyinstaller/pyinstaller.

[58] M. Mohri, A. Rostamizadeh e A. Talwalkar, Foundations of machine learning, T. Dietterich, Ed., The MIT Press, 2012.

[59] H. Debar, M. Dacier e A. Wespi, "Towards a taxonomy of intrusion-detection systems," *Computer Networks,* vol. 31, nº 8, pp. 805-822, 1999.

[60] H. Jiawei e M. Kamber, "Data mining: concepts and techniques," *San Francisco, CA, itd: Morgan Kaufmann,* vol. 5, 2001.

[61] J. Kirk, "This terrifying malware destroys your PC if detected," 5 5 2015. [Online]. Available: http://www.pcworld.com/article/2918632/rombertik-malware-destroys-computers-if-detected.html.

[62] T. Pang-Ning, M. Steinbach, V. Kumar e others, "Introduction to data mining," em *Library of Congress,* 2006.

Anexos

Anexo A – *tweetsniff.py*

```
import argparse
import errno
import ConfigParser
import json
import logging
import logging.handlers
import os
import re
import signal
import sys
import time

try:
	import twitter
except:
	print "[ERROR]: python-twitter is required. See https://github.com/bear/python-twitter"

try:
	import syslog
except:
	print "[INFO]: No Syslog support, logging to console"
from datetime import datetime
from dateutil import parser
from dateutil import tz
from elasticsearch import Elasticsearch
from termcolor import colored

api = None
logger = None

# Default configuration
config = {
	'statusFile': '/var/run/tweetsniff.status',
	'esServer': '',
	'keywords': '',
	'regex': '',
	'highlightColor': 'red',
	'keywordColor': 'blue',
	'cefServer': '',
	'cefPort': ''
}

def sigHandler(s, f):

	"""Cleanup once CTRL-C is received"""

	print "Killed."
	sys.exit(0)

def writeLog(msg):
```

```
        """Output a message to the console/Syslog depending on the host"""

        if os.name == "posix":
        syslog.openlog(logoption=syslog.LOG_PID,facility=syslog.LOG_MAIL)
        syslog.syslog(msg)
        else:
                print msg
    return

def writeCEFEvent(tweet):

        """Send a CEF event to a Syslog destination"""

        # print "[Debug]: Writing CEF: %s" % tweet
        cefmsg = ' CEF:0|blog.rootshell.be|tweetsniff|1.0|TwitterMsg|Received Twitter
Message|0|cs1Label=TweetHandle cs1=%s cs2Label=TweetTime cs2=%s msg=%s' %
(tweet.user.screen_name, tweet.created_at, tweet.text)
        logger.info(cefmsg)
        return

def time2Local(s):

        """Convert a 'created_at' date (UTC) to local time"""

        if not s:
                utc = datetime.utcnow()
        else:
                utc = datetime.strptime(parser.parse(s).strftime('%Y-%m-%d %H:%M:%S'), '%Y-
%m-%d %H:%M:%S')

        from_zone = tz.tzutc()
        to_zone = tz.tzlocal()
        utc = utc.replace(tzinfo=from_zone)
        return(utc.astimezone(to_zone))

def indexEs(tweet):

        """Index a new Tweet in Elasticsearch"""

        doc = tweet.AsDict()
        # Delete 'retweeted_status' - to be fixed later
        if 'retweeted_status' in doc:
                del doc['retweeted_status']
        # Delete 'urls' - to be fixed later?
        if 'urls' in doc:
                del doc['urls']

        # To fix: support different timezones? (+00:00
        try:
                doc['@timestamp'] = parser.parse(doc['created_at']).strftime("%Y-%m-
%dT%H:%M:%S+00:00")
                res = es.index(index=esIndex, doc_type='tweet', body=doc)
    except:
         print "[Warning] Can't connect to %s" % config['esServer']

        return

def updateTimeline(timeline_id):

        """Get new Tweets from twitter.com"""
```

```
	try:
		timeline = api.GetHomeTimeline(since_id=timeline_id)
	except twitter.error.TwitterError as e:
		print "[Error] Twitter returned: %s" % (e)
		return timeline_id

	if not timeline:
		return timeline_id

	last_id = timeline_id
	for t in reversed(timeline):
		text = t.text
		for r in config['regex']:
			if r:
				if re.search('('+r+')', text, re.I):
					text = text.replace(r, colored(r, config['highlightColor']))

		print "%s | %15s | %s" % (time2Local(t.created_at).strftime("%H:%M:%S"),
					t.user.screen_name.encode("utf-8"),
					text.encode("utf-8"))
		if es:
			indexEs(t)

		if logger:
			writeCEFEvent(t)

		if (long(t.id) > long(last_id)):
			last_id = t.id
	return(last_id)

def updateSearch(search_id):

	"""Get new Tweets containing specific keywords"""

	last_id = search_id
	for keyword in config['keywords']:
		if not keyword:
			continue
		try:
			tweets = api.GetSearch(term=keyword, since_id=search_id)
		except twitter.error.TwitterError as e:
			print "[Error] Twitter returned: %s" % (e)
			return(search_id)

		if not tweets:
			continue

		for t in reversed(tweets):
			text = t.text

			# Highlight keyword
			if re.search('('+keyword+')', text, re.I):
				text = text.replace(keyword, colored(keyword,
config['keywordColor']))

			for r in config['regex']:
				if r:
					if re.search('('+r+')', text, re.I):
```

```
                                        text = text.replace(r, colored(r,
config['highlightColor']))

                        print "%s | %15s | %s" %
(time2Local(t.created_at).strftime("%H:%M:%S"),
                                        t.user.screen_name.encode("utf-8"),
                                        text.encode("utf-8"))
                        if es:
                                indexEs(t)

                        if logger:
                                writeCEFEvent(t)

                        if long(t.id) > long(last_id):
                                last_id = t.id
        print "[DEBUG] last_id = %s" % last_id
        return(last_id)

def main():
        global api
        global config
        global es
        global esIndex
        global logger

        signal.signal(signal.SIGINT, sigHandler)

        parser = argparse.ArgumentParser(
                description='Display a Tweet feed')
        parser.add_argument('-c', '--config',
                dest = 'configFile',
                help = 'configuration file (default: /etc/tweetsniff.conf)',
                metavar = 'CONFIG')
        args = parser.parse_args()

        if not args.configFile:
                args.configFile = '/etc/tweetsniff.conf'

        try:
                c = ConfigParser.ConfigParser()
                c.read(args.configFile)
                # Twitter config
                consumerKey = c.get('twitterapi', 'consumer_key')
                consumerSecret = c.get('twitterapi', 'consumer_secret')
                accessTokenKey = c.get('twitterapi', 'access_token_key')
                accessTokenSecret = c.get('twitterapi', 'access_token_secret')
                config['statusFile'] = c.get('twitterapi', 'status_file')
                #Highligts
                config['highlightColor'] = c.get('highlight', 'color')
                highlightRegex = c.get('highlight', 'regex')
                # Search
                searchKeywords = c.get('search', 'keywords')
                config['keywordColor'] = c.get('search', 'color')
                # Elasticsearch config (optional)
                config['esServer'] = c.get('elasticsearch', 'server')
                esIndex = c.get('elasticsearch', 'index')
                # CEF confit
                try:
                        config['cefServer'] = c.get('cef', 'server')
                        config['cefPort'] = c.get('cef', 'port')
```

```
                except:
                        pass
        except OSError as e:
                writeLog('Cannot read config file %s: %s' % (args.configFile, e.errno()))
                exit

        print "DEBUG: %s, %s, %s, %s" %
(consumerKey,consumerSecret,accessTokenKey,accessTokenSecret)
        print "DEBUG: Regex: %s" % highlightRegex

        if searchKeywords:
                config['keywords'] = searchKeywords.split('\n')
                print "DEBUG: keywords = %s" % config['keywords']

        if highlightRegex:
                config['regex'] = highlightRegex.split('\n')

        try:
                api = twitter.Api(consumer_key = consumerKey,
                        consumer_secret = consumerSecret,
                        access_token_key = accessTokenKey,
                        access_token_secret = accessTokenSecret)
        except:
                print "[Error] Can't connect to twitter.com"
                sys.exit(1)

        if config['esServer']:
                try:
                        es = Elasticsearch(
                                [config['esServer']]
                                )
                except:
                        print "[Warning] Can't connect to %s" % config['esServer']

        if config['cefServer']:
                try:
                        logger = logging.getLogger('tweetsniff')
                        logger.setLevel(logging.INFO)
                        if config['cefPort']:
                                handler =
logging.handlers.SysLogHandler(address=(config['cefServer'], int(config['cefPort'])))
                        else:
                                handler =
logging.handlers.SysLogHandler(address=(config['cefServer'], 514))
                        logger.addHandler(handler)
                except:
                        print "[Warning] Can't configure CEF destination: %s:%s",
(config['cefServer'],config['cefPort'])

        if not os.path.isfile(config['statusFile']):
                print "DEBUG: Status file not found, starting new feed"
                timeline_id = 0
                search_id = 0

        else:
                fd = open(config['statusFile'], 'r')
          data = fd.read().split(',')
                timeline_id = data[0]
                search_id = data[1]
          fd.close()
```

```
                print "DEBUG: Restarting feed from ID %s/%s" % (timeline_id, search_id)

        while 1:
                try:
                        timeline_id = updateTimeline(timeline_id)
                        search_id = updateSearch(search_id)
                except AttributeError:
                        print "[Error] Can't connect to twitter.com"
                        sys.exit(1)

                fd = open(config['statusFile'], 'w')
                fd.write("%s,%s" % (str(timeline_id), str(search_id)))
                fd.close()
                try:
                        sleep_home = api.GetAverageSleepTime('statuses/home_timeline')
                        sleep_search = api.GetAverageSleepTime('search/tweets')
                except twitter.error.TwitterError as e:
                        print "[Error] Twitter returned: %s" % (e)

                print "DEBUG: Sleep = %s / %s" % (sleep_home, sleep_search)
                if sleep_search > sleep_home:
                        time.sleep(sleep_search)
                else:
                        time.sleep(sleep_home)

if __name__ == '__main__':
        main()
```

Anexo B – *outstream_twitter.py*

```
import dateutil
from elasticsearch import Elasticsearch
from elasticsearch import helpers
import time
import logging
import logging.handlers

def _change_doc_index(hits, index):
	for h in hits:
		h['_index'] = index
		yield h

def _deletion_set(hits, index):
	for h in hits:
		h['_index'] = index
		yield {'_op_type': 'delete', '_index': h['_index'], '_type': h['_type'], '_id': h['_id'], }

def printActions(act):
	for a in act:
		print a

def printSRC(src):
	for i in src['hits']['hits']:
		print i['_id'] + ' ' + i['_source']['text'] + ' by ' + i['_source']['user']['screen_name']

def sendCEF(hits, logger):
	for h in hits:
		cefmsg = ' CEF:0|Twitter|Twitter_unscored|1.0|TwitterMsg|Received Twitter Message|0|cs1Label=TweetHandle cs1=%s cs2Label=TweetTime cs2=%s msg=%s' % (h['_source']['user']['screen_name'], h['_source']['created_at'], h['_source']['text'])
		logger.info(cefmsg)
	return

try:
	logger = logging.getLogger('tweetsniff')
	logger.setLevel(logging.INFO)
	handler = logging.handlers.SysLogHandler(address=("10.0.10.169", 51405))
	logger.addHandler(handler)
except:
	print "[ERROR] Couldn't initialize CEF logging"

es = Elasticsearch(['127.0.0.1'])

src = es.search(index=['twitter'],scroll='60s',search_type='scan',size=10,body={ "query" : { "match_all" : {} } })
sid = src['_scroll_id']

n=0

while(1):
	src = es.scroll(scroll_id=sid, scroll='60s')
	sid = src['_scroll_id']

	resT, resF = helpers.bulk(es, _change_doc_index(src['hits']['hits'], "twitter_processed"), chunk_size=50, stats_only=True)
	sendCEF(src['hits']['hits'], logger)
```

```
        resTD, resFD = helpers.bulk(es, _deletion_set(src['hits']['hits'], "twitter"), chunk_size=50, stats_only=True)

        if(len(src['hits']['hits'])<=0):
                time.sleep(10)
                src = es.search(index=['twitter'],scroll='60s',search_type='scan',size=10,body={ "query" : { "match_all" : {} } })
                sid = src['_scroll_id']

        n+=1
```

Anexo C – *trainspotter.py*

```
import dateutil
import sys
import re
from elasticsearch import Elasticsearch
from elasticsearch import helpers
import time
from collections import defaultdict
import rawes
import logging
import logging.handlers

#UTILITY FUNCTIONS
def _change_doc_index(hits, index):
	for h in hits:
		h['_index'] = index
		yield h

def _deletion_set(hits, index):
	for h in hits:
		h['_index'] = index
		yield {'_op_type': 'delete', '_index': h['_index'], '_type': h['_type'], '_id': h['_id'], }

def sendCEF(hits, logger):
	for h in hits:
		cefmsg = ' CEF:0|Twitter|Twitter|1.0|TwitterMsg|Received Twitter Message|0|cs1Label=TweetHandle cs1=%s cs2Label=TweetTime cs2=%s msg=%s' % (h['_source']['user']['screen_name'], h['_source']['created_at'], h['_source']['text'])
		logger.info(cefmsg)
	return
#END
#CEF LOGGING

try:
	logger = logging.getLogger('tweetscored')
	logger.setLevel(logging.INFO)
	handler = logging.handlers.SysLogHandler(address=("10.0.10.169", 51405))
	logger.addHandler(handler)
except:
	print "[ERROR] Couldn't initialize CEF logging"

#END
#GET KEYWORDS#
es = Elasticsearch(['127.0.0.1'])

src = es.search(index=['serch_kws'],scroll='60s',search_type='scan',size=10,body={ "query" : { "match_all" : {} } })
sid = src['_scroll_id']
kws = src['hits']['hits']
n=0

while(1):
	src = es.scroll(scroll_id=sid, scroll='60s')
	sid = src['_scroll_id']

	if(len(src['hits']['hits'])<=0):
		break
```

```
            kws.extend(src['hits']['hits'])
            n+=1
#GOT KEYWORDS#
#SIMPLIFY LIST INTO DICT#
kw_dict = dict((d['_source']['keyword'], d['_source']['weight']) for (i, d) in enumerate(kws))
custom_query={
  "query": {
    "function_score": {
      "query": {
        "match_all": {}
      },
      "functions": [
        {
          "script_score": {
            "lang": "groovy",
            "script": "def sc=0;def msg=_source.text.findAll(/\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}|[\\w']+/);for(i in msg)if(myMap.containsKey(i))sc+=myMap[i];return sc;",
            "params":{"myMap":{}}
          }
        }
      ],
      "boost_mode":"replace"
    }
  }
}
custom_query['query']['function_score']['functions'][0]['script_score']['params']['myMap']=kw_dict
#DO SCORING

es = rawes.Elastic('localhost:9200')
src = es.get('twitter_processed/tweet/_search?scroll=1m', data=custom_query)
sid = src['_scroll_id']
string = ''

kws = src['hits']['hits']
top_num = 0.05*src['hits']['total']
entries=len(src['hits']['hits'])

#GO PULL FROM LOW-LEVEL CLIENT
while(1):
            print str(entries) + '/' + str(top_num)
            string='_search/scroll?scroll=1m&scroll_id='+sid
            src = es.get(string, data=custom_query)
            sid = src['_scroll_id']

            if(len(src['hits']['hits'])<=0):
                        break

            kws.extend(src['hits']['hits'])
            entries+=len(src['hits']['hits'])


sendCEF(kws[:int(top_num)], logger)

es = Elasticsearch(['127.0.0.1'])
resT, resF = helpers.bulk(es, _change_doc_index(kws, "twitter_scored"), chunk_size=50, stats_only=True)
resTD, resFD = helpers.bulk(es, _deletion_set(kws, "twitter_processed"), chunk_size=50, stats_only=True)
```

Anexo D – *voteup.py*

```
import dateutil
import sys
import re
from elasticsearch import Elasticsearch
from elasticsearch import helpers
import time
from collections import defaultdict

kw_dict=None

def updateList(es, keyword, new, multiplier):
	weight=0
	if new:
		if kw_dict[keyword]['_source']['type']=='derived':
			weight = kw_dict[keyword]['_source']['weight']+(2*multiplier)
		else:
			weight = kw_dict[keyword]['_source']['weight']+(5*multiplier)
		es.update(index='serch_kws', doc_type='keyword', id=kw_dict[keyword]['_id'],
body={"doc":{"type": kw_dict[keyword]['_source']['type'], "weight": weight, "keyword": keyword}})
	else:
		es.create(index='serch_kws', doc_type='keyword', body={"type": "derived", "weight":
2, "keyword": keyword})

#GET KEYWORDS#

es = Elasticsearch(['127.0.0.1'])

src = es.search(index=['serch_kws'],scroll='60s',search_type='scan',size=10,body={ "query" : {
"match_all" : {} } })
sid = src['_scroll_id']
kws = src['hits']['hits']
n=0

while(1):
	src = es.scroll(scroll_id=sid, scroll='60s')
	sid = src['_scroll_id']

	if(len(src['hits']['hits'])<=0):
		break

	kws.extend(src['hits']['hits'])
	n+=1
#GOT KEYWORDS#
#SIMPLIFY LIST INTO DICT#
kw_dict = dict((d['_source']['keyword'], dict(d, index=i)) for (i, d) in enumerate(kws))

#GET TOKENIZED MESSAGE#
message=re.findall("\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}|[\w']+",sys.argv[1]);

words = defaultdict(int)
for w in message:
	if(len(w)>=4):
		words[w] += 1
```

```
for i in words:
        if i in kw_dict:
                updateList(es, i, 1, words[i]);
        else:
                updateList(es, i, 0, words[i]);
```

Anexo E – *tweetsniff.conf*

```
[twitterapi]
consumer_key: wkgv8jpX4RQ2Rbagp8TpUfwTi
consumer_secret: oPYmvXwK7sPhYNyHIOXgINUODmef0JxIyhwnhBYolUGCv6ERxM
access_token_key: 2982650188-v9iODiaRpo5owS3FyUMYag2hnmKQHdHWXVEdfpm
access_token_secret: kgLtnl7bgtEj8iil6upsqZKMwBQ31ZVW1qlFdWB3Etpxy
status_file: ./tweetsniff.status

[search]
color: yellow
keywords: apache
        injection
        barracuda
        linux
        blue coat
        malware
        botnet
        microsoft
        breach
        overflow
        buffer
        security
        cisco
        shellcode
        cross site
        threat
        cybersecurity
        tomcat
        db2
        vuln
        ddos
        vulnerability
        denial of service
        webapps
        dos
        windows
        exploit
        wordpress
        hack
        Xss

[elasticsearch]
server: http://127.0.0.1:9200
index: twitter
```